**Online Food Ordering System**

Title: All Project Code

Course: Web Engineering Sessional + Project

Date of Submission:

Submitted To: Submitted By:

Subir Saha Towhidul Islam

Assistant Professor Level 3, Term 2

Dept. of CSE, PUST. ID: 170132

**App: (events, expections & middleware declaration)**

**>>app/Console:**

**Console: Kernel.php (** Contains the artisan commands necessary for Laravel , automatically generated with larval)

<?php

namespace App\Console;

use Illuminate\Console\Scheduling\Schedule;

use Illuminate\Foundation\Console\Kernel as ConsoleKernel;

class Kernel extends ConsoleKernel

{

    /\*\*

     \* The Artisan commands provided by your application.

     \*

     \* @var array

     \*/

    protected $commands = [// ];

    /\*\*

     \* Define the application's command schedule.

     \*

     \* @param  \Illuminate\Console\Scheduling\Schedule  $schedule

     \* @return void

     \*/

    protected function schedule(Schedule $schedule)

    {

        // $schedule->command('inspire')

        //          ->hourly();

    }

    /\*\*

     \* Register the commands for the application.

     \*

     \* @return void

     \*/

    protected function commands()

    {

        $this->load(\_\_DIR\_\_.'/Commands');

        require base\_path('routes/console.php');

    }

}

**>>app/Exception:**

**Exceptions: Handler.php (** Contains the exception handler for the application )

<?php

namespace App\Exceptions;

use Exception;

use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;

use Illuminate\Auth\AuthenticationException;

class Handler extends ExceptionHandler

{

    /\*\*

     \* A list of the exception types that are not reported.

     \*

     \* @var array

     \*/

    protected $dontReport = [

        //

    ];

    /\*\*

     \* A list of the inputs that are never flashed for validation exceptions.

     \*

     \* @var array

     \*/

    protected $dontFlash = [

        'password',

        'password\_confirmation',

    ];

    /\*\*

     \* Report or log an exception.

     \*

     \* @param  \Exception  $exception

     \* @return void

     \*/

    public function report(Exception $exception)

    {

        parent::report($exception);

    }

    /\*\*

     \* Render an exception into an HTTP response.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \Exception  $exception

     \* @return \Illuminate\Http\Response

     \*/

    public function render($request, Exception $exception)

    {

        return parent::render($request, $exception);

    }

    protected function unauthenticated($request, AuthenticationException $exception)

    {

        if ($request->expectsJson()) {

            return response()->json(['error' => 'Unauthenticated.'], 401);

        }

        return redirect()->guest(route('login'));

    }

}

**>>app/Http**

**Http:** It has sub folders for controllers, middleware and application requests.

Due to MVC Model it has model, controllers and views.

**Controllers-> Admin:** to handle route file requests, using controller class, which can group related requests handling logic, in this case, we have used admin class which will handle all request related to admins such as categories control, dashboard control, Index control, Main admin control, Menu control, Order control, Restaurant control, Settings control, Types control, User control etc.

AdminController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class AdminController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

    }

    public function index()

    {

        return view('admin.pages.dashboard');

    }

    public function profile()

    {

        return view('admin.pages.profile');

    }

    public function updateProfile(Request $request)

    {

        $user = User::findOrFail(Auth::user()->id);

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'first\_name' => 'required',

                'last\_name' => 'required',

                'email' => 'required|email|max:75|unique:users,id',

                'image\_icon' => 'mimes:jpg,jpeg,gif,png'

                 );

         $validator = \Validator::make($data,$rule);

            if ($validator->fails())

            {

                    return redirect()->back()->withErrors($validator->messages());

            }

        $inputs = $request->all();

        $icon = $request->file('user\_icon');

        /\*if($icon){

             $filename  = str\_slug($inputs['name'], '-').'-'.time().'.'.$icon->getClientOriginalExtension();

             $path = public\_path('upload/members/');

             $icon->move($path,$filename);

             $user->image\_icon = 'upload/members/' . $filename;

           // $user->image\_icon = $hardPath;

        }\*/

        if($icon){

            $tmpFilePath = 'upload/members/';

            $hardPath =  str\_slug($inputs['first\_name'], '-').'-'.md5(time());

            $img = Image::make($icon);

            $img->fit(200, 200)->save($tmpFilePath.$hardPath.'-b.jpg');

            $img->fit(80, 80)->save($tmpFilePath.$hardPath. '-s.jpg');

            $user->image\_icon = $hardPath;

        }

        $user->first\_name = $inputs['first\_name'];

        $user->last\_name = $inputs['last\_name'];

        $user->email = $inputs['email'];

       // $user->fill($input)->save();

        $user->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

    public function updatePassword(Request $request)

    {

            //$user = User::findOrFail(Auth::user()->id);

            $data =  \Input::except(array('\_token')) ;

            $rule  =  array(

                    'password'       => 'required|confirmed',

                    'password\_confirmation'       => 'required'

                ) ;

            $validator = \Validator::make($data,$rule);

            if ($validator->fails())

            {

                    return redirect()->back()->withErrors($validator->messages());

            }

            /\* $val=$this->validate($request, [

                    'password' => 'required|confirmed',

            ]);  \*/

        $credentials = $request->only('password', 'password\_confirmation'

            );

        $user = \Auth::user();

        $user->password = bcrypt($credentials['password']);

        $user->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

}

CategoriesController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Categories;

use App\Restaurants;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class CategoriesController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

        parent::\_\_construct();

    }

    public function categories($id)    {

        $categories = Categories::where("restaurant\_id", $id)->orderBy('category\_name')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $restaurant\_id=$id;

        return view('admin.pages.categories',compact('categories','restaurant\_id'));

    }

    public function addeditCategory($id)    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $restaurant\_id=$id;

        return view('admin.pages.addeditCategory',compact('restaurant\_id'));

    }

    public function addnew(Request $request)

    {

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'category\_name' => 'required'

                 );

         $validator = \Validator::make($data,$rule);

        if ($validator->fails())

        {

                return redirect()->back()->withErrors($validator->messages());

        }

        $inputs = $request->all();

        if(!empty($inputs['id'])){

            $cat = Categories::findOrFail($inputs['id']);

        }else{

            $cat = new Categories;

        }

        $cat->restaurant\_id = $inputs['restaurant\_id'];

        $cat->category\_name = $inputs['category\_name'];

        $cat->save();

        if(!empty($inputs['id'])){

            \Session::flash('flash\_message', 'Changes Saved');

            return \Redirect::back();

        }else{

            \Session::flash('flash\_message', 'Added');

            return \Redirect::back();

        }

    }

    public function editCategory($id,$cat\_id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $cat = Categories::findOrFail($cat\_id);

          $restaurant\_id=$id;

          return view('admin.pages.addeditCategory',compact('cat','restaurant\_id'));

    }

    public function delete($id,$cat\_id)

    {

        if(Auth::User()->usertype=="Admin" or Auth::User()->usertype=="Owner")

        {

        $cat = Categories::findOrFail($cat\_id);

        $cat->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

        }

        else

        {

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

    }

    public function owner\_categories()    {

         $user\_id=Auth::User()->id;

         $restaurant= Restaurants::where('user\_id',$user\_id)->first();

         $restaurant\_id=$restaurant['id'];

         $categories = Categories::where("restaurant\_id", $restaurant\_id)->orderBy('category\_name')->get();

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.owner.categories',compact('categories','restaurant\_id'));

    }

    public function owner\_addeditCategory()    {

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

         $user\_id=Auth::User()->id;

         $restaurant= Restaurants::where('user\_id',$user\_id)->first();

         $restaurant\_id=$restaurant['id'];

         if($restaurant\_id==""){

            \Session::flash('flash\_message', 'Add Restaurant');

            return redirect('admin/myrestaurants');

        }

        return view('admin.pages.owner.addeditCategory',compact('restaurant\_id'));

    }

}

DashboardController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Types;

use App\Restaurants;

use App\Categories;

use App\Menu;

use App\Review;

use App\Order;

use App\Http\Requests;

use Illuminate\Http\Request;

class DashboardController extends MainAdminController

{

  public function \_\_construct()

    {

     $this->middleware('auth');

    }

    public function index()

    {

       if(Auth::user()->usertype=='Admin')

          {

            $types = Types::count();

            $restaurants\_count = Restaurants::count();

            $order = Order::count();

            $users = User::where('usertype', 'User')->count();

            return view('admin.pages.dashboard',compact('types','restaurants\_count','order','users'));

        }

          if(Auth::user()->usertype=='Owner')

          {

            $user\_id=Auth::User()->id;

            $restaurant= Restaurants::where('user\_id',$user\_id)->first();

            $restaurant\_id=$restaurant['id'];

            $categories\_count = Categories::where(['restaurant\_id' => $restaurant\_id])->count();

            $menu\_count = Menu::where(['restaurant\_id' => $restaurant\_id])->count();

            $order\_count = Order::where(['restaurant\_id' => $restaurant\_id])->count();

            $review\_count = Review::where(['restaurant\_id' => $restaurant\_id])->count();

               return view('admin.pages.owner\_dashboard',compact('categories\_count','menu\_count','order\_count','review\_count'));

          }

    }

}

Index Controller

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\Http\Requests;

use Illuminate\Http\Request;

class IndexController extends MainAdminController

{

    public function index()

    {

        if (Auth::check()) {

            return redirect('admin/dashboard');

        }

        return view('admin.index');

    }

    /\*\*

     \* Do user login

     \* @return $this|\Illuminate\Http\RedirectResponse

     \*/

    public function postLogin(Request $request)

    {

    //echo bcrypt('123456');

    //exit;

      $this->validate($request, [

            'email' => 'required|email', 'password' => 'required',

        ]);

        $credentials = $request->only('email', 'password');

         if (Auth::attempt($credentials, $request->has('remember'))) {

            if(Auth::user()->usertype=='banned'){

                \Auth::logout();

                return array("errors" => 'You account has been banned!');

            }

            return $this->handleUserWasAuthenticated($request);

        }

       // return array("errors" => 'The email or the password is invalid. Please try again.');

        //return redirect('/admin');

       return redirect('/admin')->withErrors('The email or the password is invalid. Please try again.');

    }

     /\*\*

     \* Send the response after the user was authenticated.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  bool  $throttles

     \* @return \Illuminate\Http\Response

     \*/

    protected function handleUserWasAuthenticated(Request $request)

    {

        if (method\_exists($this, 'authenticated')) {

            return $this->authenticated($request, Auth::user());

        }

        return redirect('admin/dashboard');

    }

    /\*\*

     \* Log the user out of the application.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function logout()

    {

        Auth::logout();

        return redirect('admin/');

    }

}

MainAdminController.php

<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;

use App\Http\Requests;

use App\Http\Controllers\Controller;

class MainAdminController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function \_\_construct()

    {

    }

}

MenuController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Menu;

use App\Categories;

use App\Restaurants;

use App\Order;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class MenuController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

        parent::\_\_construct();

    }

    public function menulist($id)    {

        $menu = Menu::where("restaurant\_id", $id)->orderBy('menu\_name')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $restaurant\_id=$id;

        return view('admin.pages.menu',compact('menu','restaurant\_id'));

    }

    public function addeditmenu($id)    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $categories = Categories::where("restaurant\_id", $id)->orderBy('category\_name')->get();

        $restaurant\_id=$id;

        return view('admin.pages.addeditmenu',compact('categories','restaurant\_id'));

    }

    public function addnew(Request $request)

    {

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'menu\_cat' => 'required',

                'menu\_name' => 'required',

                'price' => 'required'

                 );

         $validator = \Validator::make($data,$rule);

        if ($validator->fails())

        {

                return redirect()->back()->withErrors($validator->messages());

        }

        $inputs = $request->all();

        if(!empty($inputs['id'])){

            $menu = Menu::findOrFail($inputs['id']);

        }else{

            $menu = new Menu;

        }

        //Logo image

        $menu\_image = $request->file('menu\_image');

        if($menu\_image){

             \File::delete(public\_path() .'/upload/menu/'.$menu->menu\_image.'-b.jpg');

            \File::delete(public\_path() .'/upload/menu/'.$menu->menu\_image.'-s.jpg');

            $tmpFilePath = 'upload/menu/';

            $hardPath = substr($inputs['menu\_name'],0,100).'\_'.time();

            $img = Image::make($menu\_image);

            $img->save($tmpFilePath.$hardPath.'-b.jpg');

            $img->fit(100, 100)->save($tmpFilePath.$hardPath. '-s.jpg');

            $menu->menu\_image = $hardPath;

        }

        $menu->restaurant\_id = $inputs['restaurant\_id'];

        $menu->menu\_cat = $inputs['menu\_cat'];

        $menu->menu\_name = $inputs['menu\_name'];

        $menu->description = $inputs['description'];

        $menu->price = $inputs['price'];

        $menu->save();

        if(!empty($inputs['id'])){

            \Session::flash('flash\_message', 'Changes Saved');

            return \Redirect::back();

        }else{

            \Session::flash('flash\_message', 'Added');

            return \Redirect::back();

        }

    }

    public function editmenu($id,$menu\_id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $menu = Menu::findOrFail($menu\_id);

          $categories = Categories::where("restaurant\_id", $id)->orderBy('category\_name')->get();

          $restaurant\_id=$id;

          return view('admin.pages.addeditmenu',compact('menu','categories','restaurant\_id'));

    }

    public function delete($menu\_id)

    {

        if(Auth::User()->usertype=="Admin" or Auth::User()->usertype=="Owner")

        {

             $order\_obj = Order::where('item\_id',$menu\_id)->delete();

             $menu = Menu::findOrFail($menu\_id);

             $menu->delete();

            \Session::flash('flash\_message', 'Deleted');

            return redirect()->back();

        }

        else

        {

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

    }

    public function owner\_menu()

    {

        $user\_id=Auth::User()->id;

        $restaurant= Restaurants::where('user\_id',$user\_id)->first();

        $restaurant\_id=$restaurant['id'];

        $menu = Menu::where("restaurant\_id", $restaurant\_id)->orderBy('menu\_name')->get();

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.owner.menu',compact('menu','restaurant\_id'));

    }

    public function owner\_addeditmenu()    {

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $user\_id=Auth::User()->id;

        $restaurant= Restaurants::where('user\_id',$user\_id)->first();

        $restaurant\_id=$restaurant['id'];

        $categories = Categories::where("restaurant\_id", $restaurant\_id)->orderBy('category\_name')->get();

        return view('admin.pages.owner.addeditmenu',compact('categories','restaurant\_id'));

    }

    public function owner\_editmenu($menu\_id)

    {

          if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $user\_id=Auth::User()->id;

        $restaurant= Restaurants::where('user\_id',$user\_id)->first();

        $restaurant\_id=$restaurant->id;

          $menu = Menu::findOrFail($menu\_id);

          $categories = Categories::where("restaurant\_id", $restaurant\_id)->orderBy('category\_name')->get();

          return view('admin.pages.owner.addeditmenu',compact('menu','categories','restaurant\_id'));

    }

}

OrderController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Order;

use App\Categories;

use App\Restaurants;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class OrderController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

        parent::\_\_construct();

    }

    public function orderlist($id)    {

        $order\_list = Order::where("restaurant\_id", $id)->orderBy('id','desc')->orderBy('created\_date','desc')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $restaurant\_id=$id;

        return view('admin.pages.order\_list',compact('order\_list','restaurant\_id'));

    }

    public function alluser\_order()    {

        $order\_list = Order::orderBy('id','desc')->orderBy('created\_date','desc')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.order\_list\_for\_all',compact('order\_list'));

    }

    public function order\_status($id,$order\_id,$status)

    {

        $order = Order::findOrFail($order\_id);

        $order->status = $status;

        $order->save();

            \Session::flash('flash\_message', 'Status change');

            return \Redirect::back();

    }

    public function delete($id,$order\_id)

    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $order = Order::findOrFail($order\_id);

        $order->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

    }

    public function owner\_orderlist()    {

         $user\_id=Auth::User()->id;

         $restaurant= Restaurants::where('user\_id',$user\_id)->first();

         $restaurant\_id=$restaurant['id'];

        $order\_list = Order::where("restaurant\_id", $restaurant\_id)->orderBy('created\_date')->get();

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.owner.order\_list',compact('order\_list','restaurant\_id'));

    }

    public function owner\_order\_status($order\_id,$status)

    {

        $order = Order::findOrFail($order\_id);

        $order->status = $status;

        $order->save();

            \Session::flash('flash\_message', 'Status change');

            return \Redirect::back();

    }

    public function owner\_delete($order\_id)

    {

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $order = Order::findOrFail($order\_id);

        $order->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

    }

}

RestaurantController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Restaurants;

use App\Categories;

use App\Menu;

use App\Order;

use App\Types;

use App\Review;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class RestaurantsController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

        parent::\_\_construct();

    }

    public function restaurants()    {

        $restaurants = Restaurants::orderBy('restaurant\_name')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.restaurants',compact('restaurants'));

    }

    public function addeditrestaurant()    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $types = Types::orderBy('type')->get();

        return view('admin.pages.addeditrestaurant',compact('types'));

    }

    public function addnew(Request $request)

    {

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'restaurant\_type' => 'required',

                'restaurant\_name' => 'required',

                'restaurant\_address' => 'required',

                'restaurant\_logo' => 'mimes:jpg,jpeg,gif,png'

                 );

         $validator = \Validator::make($data,$rule);

        if ($validator->fails())

        {

                return redirect()->back()->withErrors($validator->messages());

        }

        $inputs = $request->all();

        if(!empty($inputs['id'])){

            $restaurant\_obj = Restaurants::findOrFail($inputs['id']);

        }else{

            $restaurant\_obj = new Restaurants;

        }

        //Slug

        if($inputs['restaurant\_slug']=="")

        {

            $restaurant\_slug = str\_slug($inputs['restaurant\_name'], "-");

        }

        else

        {

            $restaurant\_slug =str\_slug($inputs['restaurant\_slug'], "-");

        }

        //Logo image

        $restaurant\_logo = $request->file('restaurant\_logo');

        if($restaurant\_logo){

             \File::delete(public\_path() .'/upload/restaurants/'.$restaurant\_obj->restaurant\_logo.'-b.jpg');

            \File::delete(public\_path() .'/upload/restaurants/'.$restaurant\_obj->restaurant\_logo.'-s.jpg');

            $tmpFilePath = 'upload/restaurants/';

            $hardPath = substr($restaurant\_slug,0,100).'\_'.time();

            $img = Image::make($restaurant\_logo);

            $img->fit(120, 120)->save($tmpFilePath.$hardPath.'-b.jpg');

            $img->fit(98, 98)->save($tmpFilePath.$hardPath. '-s.jpg');

            $restaurant\_obj->restaurant\_logo = $hardPath;

        }

        $user\_id=Auth::User()->id;

        $restaurant\_obj->user\_id = $user\_id;

        $restaurant\_obj->restaurant\_type = $inputs['restaurant\_type'];

        $restaurant\_obj->restaurant\_name = $inputs['restaurant\_name'];

        $restaurant\_obj->restaurant\_slug = $restaurant\_slug;

        $restaurant\_obj->restaurant\_address = $inputs['restaurant\_address'];

        $restaurant\_obj->restaurant\_description = $inputs['restaurant\_description'];

        $restaurant\_obj->open\_monday = $inputs['open\_monday'];

        $restaurant\_obj->open\_tuesday = $inputs['open\_tuesday'];

        $restaurant\_obj->open\_wednesday = $inputs['open\_wednesday'];

        $restaurant\_obj->open\_thursday = $inputs['open\_thursday'];

        $restaurant\_obj->open\_friday = $inputs['open\_friday'];

        $restaurant\_obj->open\_saturday = $inputs['open\_saturday'];

        $restaurant\_obj->open\_sunday = $inputs['open\_sunday'];

        $restaurant\_obj->save();

        if(!empty($inputs['id'])){

            \Session::flash('flash\_message', 'Changes Saved');

            return \Redirect::back();

        }else{

            \Session::flash('flash\_message', 'Added');

            return \Redirect::back();

        }

    }

    public function editrestaurant($id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $types = Types::orderBy('type')->get();

          $restaurant= Restaurants::findOrFail($id);

          return view('admin.pages.addeditrestaurant',compact('restaurant','types'));

    }

    public function delete($id)

    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $menu\_obj = Menu::where(['restaurant\_id' => $id])->delete();

        $review\_obj = Review::where('restaurant\_id',$id)->delete();

        $order\_obj = Order::where('restaurant\_id',$id)->delete();

        $cat = Restaurants::findOrFail($id);

        $cat->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

    }

    public function restaurantview($id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $restaurant= Restaurants::findOrFail($id);

          $categories\_count = Categories::where(['restaurant\_id' => $id])->count();

          $menu\_count = Menu::where(['restaurant\_id' => $id])->count();

          $order\_count = Order::where(['restaurant\_id' => $id])->count();

          $review\_count = Review::where(['restaurant\_id' => $id])->count();

          return view('admin.pages.restaurantview',compact('restaurant','categories\_count','menu\_count','order\_count','review\_count'));

    }

    public function reviewlist($id)    {

        $review\_list = Review::where("restaurant\_id", $id)->orderBy('date')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $restaurant\_id=$id;

        return view('admin.pages.review\_list',compact('review\_list','restaurant\_id'));

    }

    public function my\_restaurants()

    {

          if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

         }

         $user\_id=Auth::User()->id;

         $restaurant= Restaurants::where('user\_id',$user\_id)->first();

          $types = Types::orderBy('type')->get();

         /\* $restaurant= Restaurants::findOrFail($id);

          $categories\_count = Categories::where(['restaurant\_id' => $id])->count();

          $menu\_count = Menu::where(['restaurant\_id' => $id])->count();

          $order\_count = Order::where(['restaurant\_id' => $id])->count();

          $review\_count = Review::where(['restaurant\_id' => $id])->count();

          return view('admin.pages.restaurantview',compact('restaurant','categories\_count','menu\_count','order\_count','review\_count'));\*/

          return view('admin.pages.owner\_restaurantview',compact('restaurant','types'));

    }

    public function owner\_reviewlist()    {

        if(Auth::User()->usertype!="Owner"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $user\_id=Auth::User()->id;

        $restaurant= Restaurants::where('user\_id',$user\_id)->first();

        $restaurant\_id=$restaurant['id'];

        $review\_list = Review::where("restaurant\_id", $restaurant\_id)->orderBy('date')->get();

        return view('admin.pages.owner.review\_list',compact('review\_list','restaurant\_id'));

    }

}

SettingsController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Settings;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class SettingsController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

    }

    public function settings()

    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $settings = Settings::findOrFail('1');

        return view('admin.pages.settings',compact('settings'));

    }

    public function settingsUpdates(Request $request)

    {

        $settings = Settings::findOrFail('1');

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'site\_name' => 'required',

                'site\_email' => 'required'

                 );

         $validator = \Validator::make($data,$rule);

            if ($validator->fails())

            {

                    return redirect()->back()->withErrors($validator->messages());

            }

        $inputs = $request->all();

        $icon = $request->file('site\_logo');

        $icon\_favicon = $request->file('site\_favicon');

        //Logo

        if($icon){

           // $icon->move(public\_path().'/upload/', 'logo.png');

            $tmpFilePath = 'upload/';

            $hardPath = 'logo.png';

            $img = Image::make($icon);

            $img->save($tmpFilePath.$hardPath);

            $settings->site\_logo = 'logo.png';

        }

        //Favicon

        if($icon\_favicon){

            //$icon\_favicon->move(public\_path().'/upload/', 'favicon.png');

            //$settings->site\_favicon = 'favicon.png';

            $tmpFilePath = 'upload/';

            $hardPath = 'favicon.png';

            $img = Image::make($icon\_favicon);

            $img->fit(16, 16)->save($tmpFilePath.$hardPath);

            $settings->site\_favicon = 'favicon.png';

        }

        $settings->site\_name = $inputs['site\_name'];

        $settings->currency\_symbol = $inputs['currency\_symbol'];

        $settings->site\_email = $inputs['site\_email'];

        $settings->site\_description = $inputs['site\_description'];

        $settings->site\_copyright = $inputs['site\_copyright'];

        $settings->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

    public function homepage\_settings(Request $request)

    {

        $settings = Settings::findOrFail('1');

        $data =  \Input::except(array('\_token')) ;

        $inputs = $request->all();

        $home\_slide\_image1 = $request->file('home\_slide\_image1');

        //Home slide 1

        if($home\_slide\_image1){

            $tmpFilePath = 'upload/';

            $hardPath = 'home\_slide\_image1.png';

            $img = Image::make($home\_slide\_image1);

            $img->save($tmpFilePath.$hardPath);

            $settings->home\_slide\_image1 = 'home\_slide\_image1.png';

        }

        $home\_slide\_image2 = $request->file('home\_slide\_image2');

        //Home slide 2

        if($home\_slide\_image2){

            $tmpFilePath = 'upload/';

            $hardPath = 'home\_slide\_image2.png';

            $img = Image::make($home\_slide\_image2);

            $img->save($tmpFilePath.$hardPath);

            $settings->home\_slide\_image2 = 'home\_slide\_image2.png';

        }

        $home\_slide\_image3 = $request->file('home\_slide\_image3');

        //Home slide 2

        if($home\_slide\_image3){

            $tmpFilePath = 'upload/';

            $hardPath = 'home\_slide\_image3.png';

            $img = Image::make($home\_slide\_image3);

            $img->save($tmpFilePath.$hardPath);

            $settings->home\_slide\_image3 = 'home\_slide\_image3.png';

        }

        $page\_bg\_image = $request->file('page\_bg\_image');

        //Home page bg image

        if($page\_bg\_image){

            $tmpFilePath = 'upload/';

            $hardPath = 'page\_bg\_image.png';

            $img = Image::make($page\_bg\_image);

            $img->save($tmpFilePath.$hardPath);

            $settings->page\_bg\_image = 'page\_bg\_image.png';

        }

        $settings->total\_restaurant = $inputs['total\_restaurant'];

        $settings->total\_people\_served = $inputs['total\_people\_served'];

        $settings->total\_registered\_users = $inputs['total\_registered\_users'];

        $settings->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

    public function addthisdisqus(Request $request)

    {

        $settings = Settings::findOrFail('1');

        $data =  \Input::except(array('\_token')) ;

        $inputs = $request->all();

        $settings->addthis\_share\_code = $inputs['addthis\_share\_code'];

        $settings->disqus\_comment\_code = $inputs['disqus\_comment\_code'];

        $settings->facebook\_comment\_code = $inputs['facebook\_comment\_code'];

        $settings->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

    public function headfootupdate(Request $request)

    {

        $settings = Settings::findOrFail('1');

        $data =  \Input::except(array('\_token')) ;

        $inputs = $request->all();

        $settings->site\_header\_code = $inputs['site\_header\_code'];

        $settings->site\_footer\_code = $inputs['site\_footer\_code'];

        $settings->save();

        Session::flash('flash\_message', 'Successfully updated!');

        return redirect()->back();

    }

}

Typescontroller.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Types;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

class TypesController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

        parent::\_\_construct();

    }

    public function types()    {

        $types = Types::orderBy('type')->get();

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.types',compact('types'));

    }

    public function addeditType()    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.addedittype');

    }

    public function addnew(Request $request)

    {

        $data =  \Input::except(array('\_token')) ;

        $rule=array(

                'type' => 'required'

                 );

         $validator = \Validator::make($data,$rule);

        if ($validator->fails())

        {

                return redirect()->back()->withErrors($validator->messages());

        }

        $inputs = $request->all();

        if(!empty($inputs['id'])){

            $type\_obj = Types::findOrFail($inputs['id']);

        }else{

            $type\_obj = new Types;

        }

        //News image

        $type\_image = $request->file('type\_image');

        if($type\_image){

             \File::delete(public\_path() .'/upload/type/'.$type\_obj->type\_image.'.jpg');

            $tmpFilePath = 'upload/type/';

            $hardPath = substr($inputs['type'],0,100).'\_'.time();

            $img = Image::make($type\_image);

            $img->fit(160, 160)->save($tmpFilePath.$hardPath.'.jpg');

            //$img->fit(98, 98)->save($tmpFilePath.$hardPath. '-s.jpg');

            $type\_obj->type\_image = $hardPath;

        }

        $type\_obj->type = $inputs['type'];

        $type\_obj->save();

        if(!empty($inputs['id'])){

            \Session::flash('flash\_message', 'Changes Saved');

            return \Redirect::back();

        }else{

            \Session::flash('flash\_message', 'Added');

            return \Redirect::back();

        }

    }

    public function editType($id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $type = Types::findOrFail($id);

          return view('admin.pages.addedittype',compact('type'));

    }

    public function delete($id)

    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $type = Types::findOrFail($id);

        $type->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

    }

}

UserController.php

<?php

namespace App\Http\Controllers\Admin;

use Auth;

use App\User;

use App\Restaurants;

use App\Categories;

use App\Menu;

use App\Order;

use App\Types;

use App\Review;

use Carbon\Carbon;

use App\Http\Requests;

use Illuminate\Http\Request;

use Session;

use Intervention\Image\Facades\Image;

use Illuminate\Support\Facades\DB;

class UsersController extends MainAdminController

{

    public function \_\_construct()

    {

         $this->middleware('auth');

         parent::\_\_construct();

    }

    public function userslist()    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $allusers = User::where('usertype', '!=', 'Admin')->orderBy('id')->get();

        return view('admin.pages.users',compact('allusers'));

    }

    public function addeditUser()    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        return view('admin.pages.addeditUser');

    }

    public function addnew(Request $request)

    {

        $data =  \Input::except(array('\_token')) ;

        $inputs = $request->all();

        if(!empty($inputs['id']))

        {

            $rule=array(

                'first\_name' => 'required',

                'last\_name' => 'required',

                'email' => 'required|email|max:75'

                 );

        }

        else

        {

            $rule=array(

                'first\_name' => 'required',

                'last\_name' => 'required',

                'email' => 'required|email|max:75|unique:users',

                'password' => 'required|min:3|max:15',

                 );

        }

         $validator = \Validator::make($data,$rule);

        if ($validator->fails())

        {

                return redirect()->back()->withErrors($validator->messages());

        }

        if(!empty($inputs['id'])){

            $user = User::findOrFail($inputs['id']);

        }else{

            $user = new User;

        }

        $user->usertype = $inputs['usertype'];

        $user->first\_name = $inputs['first\_name'];

        $user->last\_name = $inputs['last\_name'];

        $user->email = $inputs['email'];

        $user->mobile = $inputs['mobile'];

        $user->city = $inputs['city'];

        $user->postal\_code = $inputs['postal\_code'];

        $user->address = $inputs['address'];

        if($inputs['password'])

        {

            $user->password= bcrypt($inputs['password']);

        }

        $user->save();

        if(!empty($inputs['id'])){

            \Session::flash('flash\_message', 'Changes Saved');

            return \Redirect::back();

        }else{

            \Session::flash('flash\_message', 'Added');

            return \Redirect::back();

        }

    }

    public function editUser($id)

    {

          if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

          $user = User::findOrFail($id);

          return view('admin.pages.addeditUser',compact('user'));

    }

    public function delete($id)

    {

        if(Auth::User()->usertype!="Admin"){

            \Session::flash('flash\_message', 'Access denied!');

            return redirect('admin/dashboard');

        }

        $review\_obj = Review::where('user\_id',$id)->delete();

        $resta\_obj = Restaurants::where('user\_id',$id)->delete();

        $order\_obj = Order::where('user\_id',$id)->delete();

        $user = User::findOrFail($id);

        \File::delete(public\_path() .'/upload/members/'.$user->image\_icon.'-b.jpg');

        \File::delete(public\_path() .'/upload/members/'.$user->image\_icon.'-s.jpg');

        $user->delete();

        \Session::flash('flash\_message', 'Deleted');

        return redirect()->back();

    }

}

**>>app/Http/Controllers/Auth:**

The main authentication file is located at ./app/cofig/auth.php

This directory contains the code for auth, which is auto generated and used the database to store user and admin infos .

It has multiple files to password control, reset password control, verification control, login controller etc. codes are given below:

ForgotPasswordController.php

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\SendsPasswordResetEmails;

class ForgotPasswordController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Password Reset Controller

    |--------------------------------------------------------------------------

    |

    | This controller is responsible for handling password reset emails and

    | includes a trait which assists in sending these notifications from

    | your application to your users. Feel free to explore this trait.

    |

    \*/

    use SendsPasswordResetEmails;

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('guest');

    }

}

LoginController.php

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Login Controller

    |--------------------------------------------------------------------------

    |

    | This controller handles authenticating users for the application and

    | redirecting them to your home screen. The controller uses a trait

    | to conveniently provide its functionality to your applications.

    |

    \*/

    use AuthenticatesUsers;

    /\*\*

     \* Where to redirect users after login.

     \*

     \* @var string

     \*/

    protected $redirectTo = '/home';

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('guest')->except('logout');

    }

}

PasswordController.php

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\ResetsPasswords;

use Illuminate\Http\Request;

use Illuminate\Mail\Message;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Password;

use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;

class PasswordController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Password Reset Controller

    |--------------------------------------------------------------------------

    |

    | This controller is responsible for handling password reset requests

    | and uses a simple trait to include this behavior. You're free to

    | explore this trait and override any methods you wish to tweak.

    |

    \*/

    use ResetsPasswords;

    /\*\*

     \* Create a new password controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('guest');

    }

    public function getEmail()

    {

        return view('admin.password');

    }

    public function postEmail(Request $request)

   {

       $this->validate($request, ['email' => 'required|email']);

       $response = Password::sendResetLink($request->only('email'), function (Message $message) {

           $message->subject('Your Password Reset Link');

           $message->sender(getcong('site\_email'));

       });

       switch ($response) {

           case Password::RESET\_LINK\_SENT:

               \Session::flash('flash\_message', 'We have e-mailed your password reset link!');

               return redirect()->back();

           case Password::INVALID\_USER:

               return redirect()->back()->withErrors(['email' => trans($response)]);

       }

   }

   /\*\*

     \* Display the password reset view for the given token.

     \*

     \* @param  string  $token

     \* @return \Illuminate\Http\Response

     \*/

    public function getReset($token = null)

    {

        if (is\_null($token)) {

            throw new NotFoundHttpException;

        }

        return view('admin.auth.reset')->with('token', $token);

    }

    /\*\*

     \* Reset the given user's password.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

     \*/

    public function postReset(Request $request)

    {

        $this->validate($request, [

            'token' => 'required',

            'email' => 'required|email',

            'password' => 'required|confirmed|min:6',

        ]);

        $credentials = $request->only(

            'email', 'password', 'password\_confirmation', 'token'

        );

        $response = Password::reset($credentials, function ($user, $password) {

            $this->resetPassword($user, $password);

        });

        switch ($response) {

            case Password::PASSWORD\_RESET:

                \Session::flash('success.message',  trans($response));

                return redirect('admin/dashboard');

            default:

                \Session::flash('error.message',  trans($response));

                return redirect()->back()

                    ->withInput($request->only('email'));

        }

    }

    /\*\*

     \* Reset the given user's password.

     \*

     \* @param  \Illuminate\Contracts\Auth\CanResetPassword  $user

     \* @param  string  $password

     \* @return void

     \*/

    protected function resetPassword($user, $password)

    {

        $user->password = bcrypt($password);

        $user->save();

        Auth::login($user);

    }

}

RegisterController.php

<?php

namespace App\Http\Controllers\Auth;

use App\User;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Register Controller

    |--------------------------------------------------------------------------

    |

    | This controller handles the registration of new users as well as their

    | validation and creation. By default this controller uses a trait to

    | provide this functionality without requiring any additional code.

    |

    \*/

    use RegistersUsers;

    /\*\*

     \* Where to redirect users after registration.

     \*

     \* @var string

     \*/

    protected $redirectTo = '/home';

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('guest');

    }

    /\*\*

     \* Get a validator for an incoming registration request.

     \*

     \* @param  array  $data

     \* @return \Illuminate\Contracts\Validation\Validator

     \*/

    protected function validator(array $data)

    {

        return Validator::make($data, [

            'name' => 'required|string|max:255',

            'email' => 'required|string|email|max:255|unique:users',

            'password' => 'required|string|min:6|confirmed',

        ]);

    }

    /\*\*

     \* Create a new user instance after a valid registration.

     \*

     \* @param  array  $data

     \* @return \App\User

     \*/

    protected function create(array $data)

    {

        return User::create([

            'name' => $data['name'],

            'email' => $data['email'],

            'password' => Hash::make($data['password']),

        ]);

    }

}

ResetPasswordController.php

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\ResetsPasswords;

class ResetPasswordController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Password Reset Controller

    |--------------------------------------------------------------------------

    |

    | This controller is responsible for handling password reset requests

    | and uses a simple trait to include this behavior. You're free to

    | explore this trait and override any methods you wish to tweak.

    |

    \*/

    use ResetsPasswords;

    /\*\*

     \* Where to redirect users after resetting their password.

     \*

     \* @var string

     \*/

    protected $redirectTo = '/home';

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('guest');

    }

}

VerificationController.php

<?php

namespace App\Http\Controllers\Auth;

use Illuminate\Http\Request;

use Illuminate\Routing\Controller;

use Illuminate\Foundation\Auth\VerifiesEmails;

class VerificationController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Email Verification Controller

    |--------------------------------------------------------------------------

    |

    | This controller is responsible for handling email verification for any

    | user that recently registered with the application. Emails may also

    | be resent if the user did not receive the original email message.

    |

    \*/

    use VerifiesEmails;

    /\*\*

     \* Where to redirect users after verification.

     \*

     \* @var string

     \*/

    protected $redirectTo = '/home';

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

        $this->middleware('auth');

        $this->middleware('signed')->only('verify');

        $this->middleware('throttle:6,1')->only('verify', 'resend');

    }

}

It also has some other files CartController.php, Controller.php, IndexController.php, RestaurantsController.php etc.

This files also serves the same purpose .

>>**app/Http/Middleware**

Http middleware provides mechanism for filtering HTTP requests entering the application. Such as middleware that verifies the users are either authenticated or not. If the user is authenticated then the middleware will redirect the user to the login screen. If not authenticated the middleware won’t let the request to be processed by the application, from this sense it may include many files. Besides middle wares can be used to perform a variety of tasks,

**>>app/Http/middleware/Authenticate.php**

<?php

namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;

class Authenticate extends Middleware

{

    /\*\*

     \* Get the path the user should be redirected to when they are not authenticated.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return string

     \*/

    protected function redirectTo($request)

    {

        return route('login');

    }

}

**>>app/Http/middleware/CheckForMaintainanceMode.php**

<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode as Middleware;

class CheckForMaintenanceMode extends Middleware

{

    /\*\*

     \* The URIs that should be reachable while maintenance mode is enabled.

     \*

     \* @var array

     \*/

    protected $except = [

        //

    ];

}

**>>app/Http/middleware/RedirectIfAuthenticated.php**

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Support\Facades\Auth;

class RedirectIfAuthenticated

{

    /\*\*

     \* Handle an incoming request.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \Closure  $next

     \* @param  string|null  $guard

     \* @return mixed

     \*/

    public function handle($request, Closure $next, $guard = null)

    {

        if (Auth::guard($guard)->check()) {

            return redirect('/home');

        }

        return $next($request);

    }

}

**This directories have another files too such as helpers.php and kernel.php.**

**>>app/Providers**

This directory contains the necessary code for bootstrapping. The initial file is found in **app/config/app.php**  in which we will find an array named **provider.** Those are service provider classes that has to be loaded for the application to run.

Those are defined here in this directory,

**>>app/Providers/AppServiceProvider.php**

<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

use Illuminate\Support\Facades\Schema;

class AppServiceProvider extends ServiceProvider

{

    /\*\*

     \* Bootstrap any application services.

     \*

     \* @return void

     \*/

    public function boot()

    {

        Schema::defaultStringLength(191);

    }

    /\*\*

     \* Register any application services.

     \*

     \* @return void

     \*/

    public function register()

    {

        //

    }

}

**>>app/Providers/AuthServiceProvider.php**

<?php

namespace App\Providers;

use Illuminate\Support\Facades\Gate;

use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;

class AuthServiceProvider extends ServiceProvider

{

    /\*\*

     \* The policy mappings for the application.

     \*

     \* @var array

     \*/

    protected $policies = [

        'App\Model' => 'App\Policies\ModelPolicy',

    ];

    /\*\*

     \* Register any authentication / authorization services.

     \*

     \* @return void

     \*/

    public function boot()

    {

        $this->registerPolicies();

        //

    }

}

**>>app/Providers/RouteServiceProvider.php**

<?php

namespace App\Providers;

use Illuminate\Support\Facades\Route;

use Illuminate\Foundation\Support\Providers\RouteServiceProvider as ServiceProvider;

class RouteServiceProvider extends ServiceProvider

{

    /\*\*

     \* This namespace is applied to your controller routes.

     \*

     \* In addition, it is set as the URL generator's root namespace.

     \*

     \* @var string

     \*/

    protected $namespace = 'App\Http\Controllers';

    /\*\*

     \* Define your route model bindings, pattern filters, etc.

     \*

     \* @return void

     \*/

    public function boot()

    {

        //

        parent::boot();

    }

    /\*\*

     \* Define the routes for the application.

     \*

     \* @return void

     \*/

    public function map()

    {

        $this->mapApiRoutes();

        $this->mapWebRoutes();

        //

    }

    /\*\*

     \* Define the "web" routes for the application.

     \*

     \* These routes all receive session state, CSRF protection, etc.

     \*

     \* @return void

     \*/

    protected function mapWebRoutes()

    {

        Route::middleware('web')

             ->namespace($this->namespace)

             ->group(base\_path('routes/web.php'));

    }

    /\*\*

     \* Define the "api" routes for the application.

     \*

     \* These routes are typically stateless.

     \*

     \* @return void

     \*/

    protected function mapApiRoutes()

    {

        Route::prefix('api')

             ->middleware('api')

             ->namespace($this->namespace)

             ->group(base\_path('routes/api.php'));

    }

}

**>>App/cart.php**

**>>app/Catrogies.php**

**>>app/Menu.php**

**>>app/Restaurants.php**

**>>app/Review.php**

**>>app/Settings.php**

**>>app/User.php**

**>>app/Types.php**

This all files includes necessary codes for new classes that will be extended and then used by the application using MVC’s model.

>>**app/config/**

This directory contains the configuration files. Which allows us to configure thigs like database connection, timezone and application key.

The **.env** file may include many variable types such as Booleans and others.

>>app/config/.env

APP\_NAME=Laravel

APP\_ENV=local

APP\_KEY=

APP\_DEBUG=true

APP\_URL=http://localhost

LOG\_CHANNEL=stack

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=food

DB\_USERNAME=root

DB\_PASSWORD=

BROADCAST\_DRIVER=log

CACHE\_DRIVER=file

QUEUE\_CONNECTION=sync

SESSION\_DRIVER=file

SESSION\_LIFETIME=120

REDIS\_HOST=127.0.0.1

REDIS\_PASSWORD=null

REDIS\_PORT=6379

MAIL\_DRIVER=smtp

MAIL\_HOST=smtp.mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

PUSHER\_APP\_ID=

PUSHER\_APP\_KEY=

PUSHER\_APP\_SECRET=

PUSHER\_APP\_CLUSTER=mt1

MIX\_PUSHER\_APP\_KEY="${PUSHER\_APP\_KEY}"

MIX\_PUSHER\_APP\_CLUSTER="${PUSHER\_APP\_CLUSTER}"

**>>app/database**

This directory contains code for migration and seeds.

**>>app/database/migrations/**

**>>app/database/seeds/**

**>>app/database/factories**

**>>app/public**