Fast Travel

Powered by Laravel

Jovan Sekulic 298/16

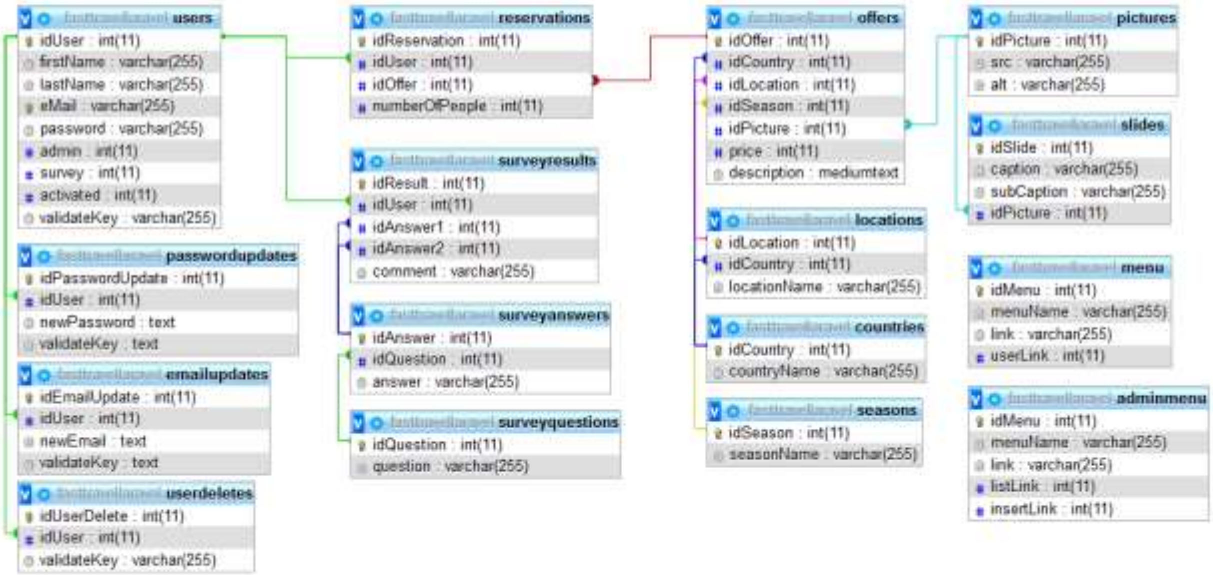
Visoka ICT Skola

Table of Contents

Aim of this website

The aim of this website is to provide the customer with infomation about travel plan offers by this fictional travel company. The user can register and log in on this website. When logged in, the user has the ability to make reservations in advance and also to cancel them if needed. The user also can take a one-time survey to share their user experience with the fictional company. Other noteworthy features of the user is to change his/her email address and password; and to delete the account. If the user doesn't want to register, he/she can view and search all available offers and send an email to the site administrator, provided by the form on the website. The site administrator has the ability to insert, edit and delete offers, slides, countries and locations, also he/she can view all of the survey results posted by registered users.

Database structure



Routes

```
Route::get('/', 'OfferController@index');
Route::get('/offers', 'OfferController@index');
Route::post('/offers', 'OfferController@search');
Route::get('/offers/{id}', 'OfferController@show');
Route::post('/filterLocations','LocationController@filter');

Route::get('/contact','ContactController@index');
Route::post('/contact','ContactController@submit');

Route::get('/survey','SurveyController@create');

Route::get('/about','PageController@about');

Route::get('/register','UserController@create');
Route::post('/register','UserController@store');
Route::get('/validate/{key}','UserController@validateUser');

Route::post('/user/login','UserController@login');

Route::get('/validatePasswordChange/{key}','PasswordUpdateController@validateUpdate');
Route::get('/validateEmailChange/{key}','EmailUpdateController@validateUpdate');
Route::get('/validateUserDelete/{key}','UserController@validateDelete');

Route::group(['middleware'=>['auth']],function (){
    Route::post('/reserve/{id}','ReservationController@store');
    Route::get('/user/reservations','ReservationController@index');
    Route::get('/user/reservations/cancel/{id}','ReservationController@destroy');

    Route::post('/survey','SurveyController@submit');

    Route::get('user/delete','UserController@delete');

    Route::post('/user/logout','UserController@logout');
    Route::get('/user/dashboard','UserController@index');
    Route::get('/user/changePassword','PasswordUpdateController@edit');
    Route::post('/user/changePassword','PasswordUpdateController@update');
    Route::get('/user/changeEmail','EmailUpdateController@edit');
    Route::post('/user/changeEmail','EmailUpdateController@update');
});

Route::group(['middleware'=>['auth.admin']],function (){
    Route::get('/admin','OfferController@indexTable');
    Route::get('/offers/delete/{id}','OfferController@destroy');
    Route::get('/admin/insert/offer','OfferController@create');
    Route::post('/admin/insert/offer','OfferController@store');
    Route::get('/offers/edit/{id}','OfferController@edit');
    Route::post('/offers/edit/{id}','OfferController@update');

    Route::get('/admin/slides','SlideController@index');
    Route::get('/slides/delete/{id}','SlideController@destroy');
    Route::get('/admin/insert/slide','SlideController@create');
    Route::post('/admin/insert/slide','SlideController@store');
    Route::get('/slides/edit/{id}','SlideController@edit');
    Route::post('/slides/edit/{id}','SlideController@update');

    Route::get('/admin/locations','LocationController@index');
    Route::get('/locations/delete/{id}','LocationController@destroy');
    Route::get('/admin/insert/location','LocationController@create');
    Route::post('/admin/insert/location','LocationController@store');
    Route::get('/locations/edit/{id}','LocationController@edit');
    Route::post('/locations/edit/{id}','LocationController@update');

    Route::get('/admin/countries','CountryController@index');
    Route::get('/countries/delete/{id}','CountryController@destroy');
    Route::get('/admin/insert/country','CountryController@create');
    Route::post('/admin/insert/country','CountryController@store');
    Route::get('/countries/edit/{id}','CountryController@edit');
    Route::post('/countries/edit/{id}','CountryController@update');

    Route::get('/admin/surveyResults','SurveyController@index');
});
```

Custom middleware used

Check if the user is logged in

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Auth;

class CheckLogin
{
    public function handle($request, Closure $next)
    {
        if(Auth::check()){
            return $next($request);
        }
        return redirect('/');
    }
}
```

Check if the user is logged in and is an admin

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Auth;

class CheckAdmin
{
    public function handle($request, Closure $next)
    {
        if(Auth::check() && Auth::user()->admin==1){
            return $next($request);
        }
        return redirect('/');
    }
}
```

Models

Users

When deleting a user, all reservations are canceled and if the user has completed the survey, it’s results are also removed.

```
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use Notifiable;

    protected $table = "users";
    protected $primaryKey = "idUser";
    public $timestamps = false;

    public static function boot()
    {
        parent::boot();
        static::deleting(function (User $toDelete){
            $surveyResults = SurveyResults::all()->where('idUser','=',$toDelete-
>idUser);
            foreach ($surveyResults as $surveyResult){
                $surveyResult->delete();
            }
            $reservations = Reservations::all()->where('idUser','=',$toDelete-
>idUser);
            foreach ($reservations as $reservation){
                $reservation->delete();
            }
        });
    }

    protected $fillable = [
        'name', 'email', 'password',
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];

    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

PasswordUpdates

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class PasswordUpdates extends Model
{
    protected $table = "passwordupdates";
    protected $primaryKey = "idPasswordUpdate";
    public $timestamps = false;

    public function user(){
        return $this->hasOne('App\User','idUser','idUser');
    }
}
```

EmailUpdates

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class EmailUpdates extends Model
{
    protected $table = "emailupdates";
    protected $primaryKey = "idEmailUpdate";
    public $timestamps = false;

    public function user(){
        return $this->hasOne('App\User','idUser','idUser');
    }
}
```

UserDeletes

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class UserDeletes extends Model
{
    protected $table = "userdeletes";
    protected $primaryKey = "idUserDelete";
    public $timestamps = false;
}
```

Survey

SurveyQuestions

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class SurveyQuestions extends Model
{
    protected $table = "surveyquestions";
    protected $primaryKey = "idQuestion";
    public $timestamps = false;
}
```

SurveyAnswers

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class SurveyAnswers extends Model
{
    protected $table = "surveyanswers";
    protected $primaryKey = "idAnswer";
    public $timestamps = false;

    public function question(){
        return $this->hasOne('App\SurveyQuestions','idQuestion','idQuestion');
    }
}
```

SurveyResults

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class SurveyResults extends Model
{
    protected $table = "surveyresults";
    protected $primaryKey = "idResult";
    public $timestamps = false;

    public function user(){
        return $this->hasOne('App\User','idUser','idUser');
    }

    public function answer1(){
        return $this->hasOne('App\SurveyAnswers','idAnswer','idAnswer1');
    }

    public function answer2(){
        return $this->hasOne('App\SurveyAnswers','idAnswer','idAnswer2');
    }
}
```

Menu

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Menus extends Model
{
    protected $table = "menu";
    protected $primaryKey = "idMenu";
    public $timestamps = false;
}
```

AdminMenu

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class AdminMenus extends Model
{
    protected $table = "adminmenu";
    protected $primaryKey = "idMenu";
    public $timestamps = false;
}
```

Pictures

While deleting, the file on the server is automatically deleted.

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\File;

class Pictures extends Model
{
    protected $table = "pictures";
    protected $primaryKey = "idPicture";
    public $timestamps = false;

    public static function boot()
    {
        parent::boot(); // TODO: Change the autogenerated stub
        static::deleted(function(Pictures $toDelete){
            File::delete(public_path().$toDelete->src);
        });
    }
}
```

Slides

While deleting, the relevant picture is deleted.

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Slides extends Model
{
    protected $table = "slides";
    protected $primaryKey = "idSlide";
    public $timestamps = false;

    public function picture(){
        return $this->hasOne('App\Pictures','idPicture','idPicture');
    }

    public static function boot(){
        parent::boot();
        static::deleted(function (Slides $toDelete){
            $picture = Pictures::find($toDelete->idPicture);
            $picture->delete();
        });
    }
}
```

Offers

Countries

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Countries extends Model
{
    protected $table = "countries";
    protected $primaryKey = "idCountry";
    public $timestamps = false;

    public static function boot()
    {
        parent::boot(); // TODO: Change the autogenerated stub
        static::deleting(function(Countries $toDelete){
            $locations = Locations::all()->where('idCountry','=',$toDelete-
>idCountry);
            if($locations->count()>0){
                foreach ($locations as $location){
                    $location->delete();
                }
            }
        });
    }
}
```

Locations

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Locations extends Model
{
    protected $table = "locations";
    protected $primaryKey = "idLocation";
    public $timestamps = false;

    public function country(){
        return $this->hasOne("App\Countries","idCountry","idCountry");
    }
}
```

Seasons

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Seasons extends Model
{
    protected $table = "seasons";
    protected $primaryKey = "idSeason";
    public $timestamps = false;
}
```

Offers

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Offers extends Model
{
    protected $table = "offers";
    protected $primaryKey = "idOffer";
    public $timestamps = false;

    public function country(){
        return $this->hasOne("App\Countries","idCountry","idCountry");
    }

    public function location(){
        return $this->hasOne("App\Locations","idLocation","idLocation");
    }

    public function season(){
        return $this->hasOne("App\Seasons","idSeason","idSeason");
    }

    public function picture(){
        return $this->hasOne('App\Pictures','idPicture','idPicture');
    }

    public static function boot()
    {
        parent::boot(); // TODO: Change the autogenerated stub
        static::deleted(function (Offers $toDelete){
            $picture = Pictures::find($toDelete->idPicture);
            $picture->delete();
        });
    }
}
```

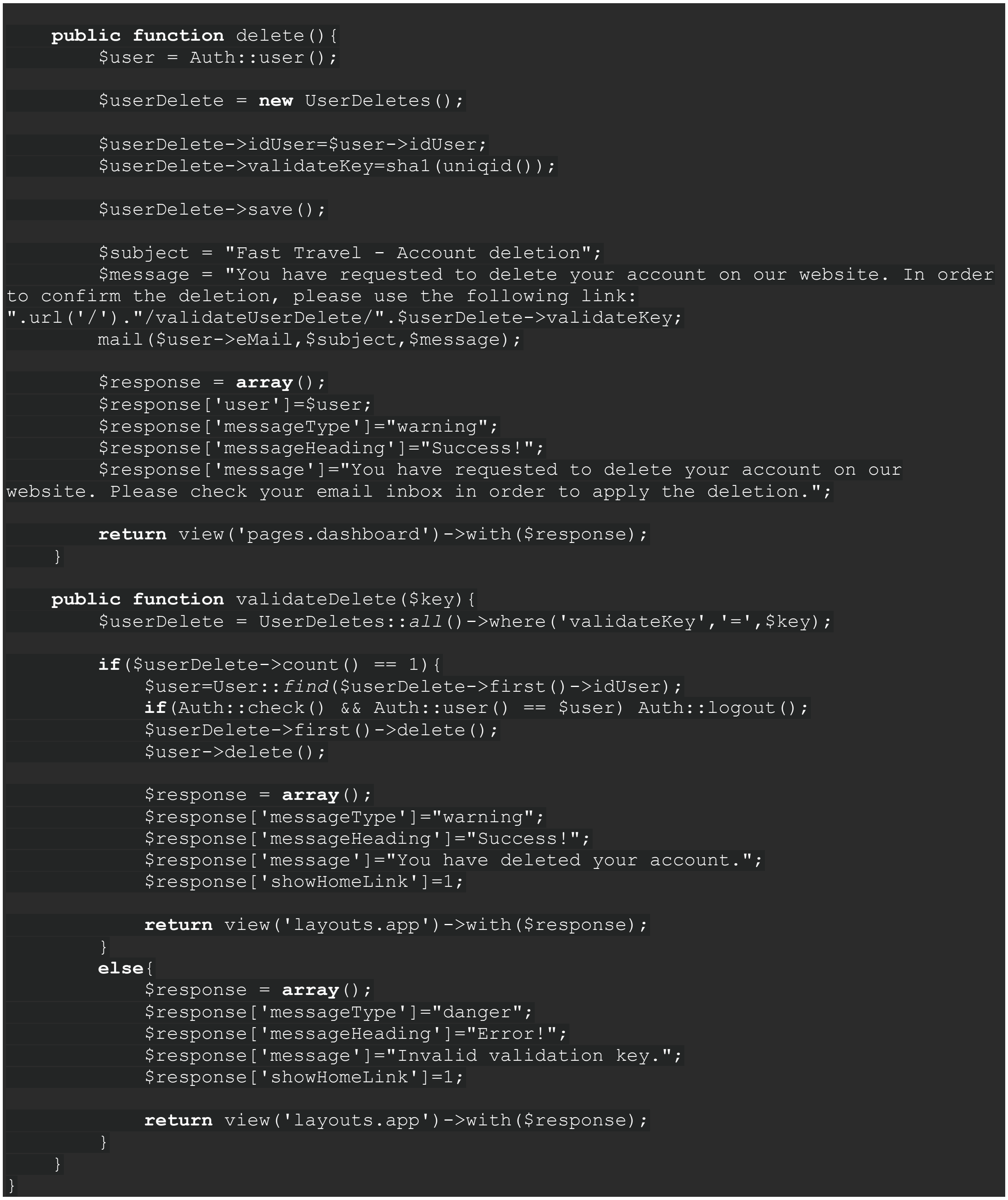
Reservations

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Reservations extends Model
{
    protected $table = "reservations";
    protected $primaryKey = "idReservation";
    public $timestamps = false;

    public function offer(){
        return $this->hasOne('App\Offers','idOffer','idOffer');
    }

    public function user(){
        return $this->hasOne('App\User','idUser','idUser');
    }
}
```

Controllers

UserController

```
<?php

namespace App\Http\Controllers;

use App\User;
use App\UserDeletes;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class UserController extends Controller
{
    public function index()
    {
        $user=Auth::user();
        return view('pages.dashboard')->with('user',$user);
    }

    public function create()
    {
        return view('pages.register');
    }

    public function store(Request $request)
    {
        $nameRegex = "/^[a-zA-
ZàáâäãåąčćęèéêëėįìíîïłńòóôöõøùúûüųūÿýżźñçčšžÀÁÂÄÃÅĄĆČĖĘÈÉÊËÌÍÎÏĮŁŃÒÓÔÖÕØÙÚÛÜŲŪŸÝŻŹÑßÇŒ
ÆČŠŽ∂ð ,.'-]+$/u";
        $emailRegex = "/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0-9\-]+\.)+[a-
z]{2,6}$/ix";
        $passwordRegex = "/^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-
9])(?=.*[!@#\$%\^&\*])(?=.{8,})/";

        $request->validate([
            'firstName' => 'required|regex:'.$nameRegex,
            'lastName' => 'required|regex:'.$nameRegex,
            'eMail' => 'required|unique:users,eMail|regex:'.$emailRegex,
            'password' => 'required|confirmed|regex:'.$passwordRegex,
        ]);

        $firstName = $request->input('firstName');
        $lastName = $request->input('lastName');
        $eMail = $request->input('eMail');
        $password = $request->input('password');

        $user = new User();

        $user->admin=0;
        $user->survey=0;
        $user->activated=0;
        $user->validateKey=sha1(uniqid());

        $user->firstName=$firstName;
        $user->lastName=$lastName;
        $user->eMail=$eMail;
        $user->password=Hash::make($password);

        $user->save();
        $subject = "Fast Travel - Account activation";
        $message = "Thank you for registering on our website. In order to activate
your account, please use the following link: ".url('/')."/validate/".$user-
>validateKey;
        mail($eMail,$subject,$message);

        $response = array();
        $response['messageType']="success";
        $response['messageHeading']="Success!";
        $response['message']="You have registered on our website. Please check your
email in order to activate your account.";

        return view('pages.register')->with($response);
    }

    public function validateUser($key){
        $user = User::all()->where('validateKey','=',$key);

        if($user->count() == 1){
            $user->first()->activated=1;
            $user->first()->validateKey="";

            $user->first()->save();

            $response = array();
            $response['messageType']="success";
            $response['messageHeading']="Success!";
            $response['message']="You have activated your account.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
        else{
            $response = array();
            $response['messageType']="danger";
            $response['messageHeading']="Error!";
            $response['message']="Invalid validation key.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
    }

    public function login(Request $request){
        $credentials = array();

        $credentials['eMail'] = $request->input('eMail');
        $credentials['password']  = $request->input('password');
        $credentials['activated'] = 1;

        if(Auth::attempt($credentials)){
            return redirect()->intended('/');
        }
        else{
            $loginError = "Incorrect login information.";
            return redirect()->back()->with('loginError',$loginError);
        }
    }

    public function logout(){
        Auth::logout();
        return redirect('/');
    }
```



```
    public function delete(){
        $user = Auth::user();

        $userDelete = new UserDeletes();

        $userDelete->idUser=$user->idUser;
        $userDelete->validateKey=sha1(uniqid());

        $userDelete->save();

        $subject = "Fast Travel - Account deletion";
        $message = "You have requested to delete your account on our website. In order
to confirm the deletion, please use the following link:
".url('/')."/validateUserDelete/".$userDelete->validateKey;
        mail($user->eMail,$subject,$message);

        $response = array();
        $response['user']=$user;
        $response['messageType']="warning";
        $response['messageHeading']="Success!";
        $response['message']="You have requested to delete your account on our
website. Please check your email inbox in order to apply the deletion.";

        return view('pages.dashboard')->with($response);
    }

    public function validateDelete($key){
        $userDelete = UserDeletes::all()->where('validateKey','=',$key);

        if($userDelete->count() == 1){
            $user=User::find($userDelete->first()->idUser);
            if(Auth::check() && Auth::user() == $user) Auth::logout();
            $userDelete->first()->delete();
            $user->delete();

            $response = array();
            $response['messageType']="warning";
            $response['messageHeading']="Success!";
            $response['message']="You have deleted your account.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
        else{
            $response = array();
            $response['messageType']="danger";
            $response['messageHeading']="Error!";
            $response['message']="Invalid validation key.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
    }
}
```

PasswordUpdateController

```
<?php

namespace App\Http\Controllers;

use App\PasswordUpdates;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class PasswordUpdateController extends Controller
{
    public function edit(){
        return view('pages.updatePassword');
    }

    public function update(Request $request){
        $passwordRegex = "/^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#\$%\^&\*])(?=.{8,})/";

        $request->validate([
            'newPassword' => 'required|confirmed|regex:'.$passwordRegex
        ]);

        $passwordUpdate = new PasswordUpdates();

        $passwordUpdate->idUser = Auth::user()->idUser;
        $passwordUpdate->newPassword = Hash::make($request->input('newPassword'));
        $passwordUpdate->validateKey = sha1(uniqid());

        $passwordUpdate->save();

        $subject = "Fast Travel - Password Update";
        $message = "You have sent a request to change your password. In order to apply the change, please use the following link: ".url('/')."/validatePasswordChange/".$passwordUpdate->validateKey;
        mail(Auth::user()->eMail,$subject,$message);

        $response = array();
        $response['messageType']="success";
        $response['messageHeading']="Success!";
        $response['message']="You have sent a request to change your password. Please check your email inbox in order to apply the change.";

        return view('pages.updatePassword')->with($response);
    }

    public function validateUpdate($key){
        $passwordUpdate = PasswordUpdates::all()->where('validateKey','=',$key);

        if($passwordUpdate->count() == 1){
            $user = User::all()->where('idUser','=',$passwordUpdate->first()->idUser);
            $user->first()->password = $passwordUpdate->first()->newPassword;
            $user->first()->save();
            $passwordUpdate->first()->delete();

            $response = array();
            $response['messageType']="success";
            $response['messageHeading']="Success!";
            $response['message']="You have successfully changed your password, you can now log in with your new password.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
        else{
            $response = array();
            $response['messageType']="danger";
            $response['messageHeading']="Error!";
            $response['message']="Invalid validation key.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
    }
}
```

EmailUpdateController

```
<?php

namespace App\Http\Controllers;

use App\EmailUpdates;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class EmailUpdateController extends Controller
{
    public function edit(){
        return view('pages.updateEmail');
    }

    public function update(Request $request){
        $emailRegex = "/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0-9\-]+\.)+[a-
z]{2,6}$/ix";

        $request->validate([
            'newEmail' => 'required|unique:users,eMail|regex:'.$emailRegex
        ]);

        $emailUpdate = new EmailUpdates();

        $emailUpdate->idUser = Auth::user()->idUser;
        $emailUpdate->newEmail = $request->input('newEmail');
        $emailUpdate->validateKey = sha1(uniqid());

        $emailUpdate->save();

        $subject = "Fast Travel - Email Update";
        $message = "You have sent a request to change your email address. In order to
apply the change, please use the following link:
".url('/')."/validateEmailChange/".$emailUpdate->validateKey;
        mail(Auth::user()->eMail,$subject,$message);

        $response = array();
        $response['messageType']="success";
        $response['messageHeading']="Success!";
        $response['message']="You have sent a request to change your email address.
Please check your old email inbox in order to apply the change.";

        return view('pages.updateEmail')->with($response);
    }
    public function validateUpdate($key){
        $emailUpdate = EmailUpdates::all()->where('validateKey','=',$key);

        if($emailUpdate->count() == 1){
            $user = User::all()->where('idUser','=',$emailUpdate->first()->idUser);
            $user->first()->eMail = $emailUpdate->first()->newEmail;
            $user->first()->save();
            $emailUpdate->first()->delete();

            $response = array();
            $response['messageType']="success";
            $response['messageHeading']="Success!";
            $response['message']="You have successfully changed your email address, 
you can now log in with your new email address.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
        else{
            $response = array();
            $response['messageType']="danger";
            $response['messageHeading']="Error!";
            $response['message']="Invalid validation key.";
            $response['showHomeLink']=1;

            return view('layouts.app')->with($response);
        }
    }
}
```

SurveyController

```
<?php

namespace App\Http\Controllers;

use App\SurveyAnswers;
use App\SurveyQuestions;
use App\SurveyResults;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class SurveyController extends Controller
{
    public function index()
    {
        $response = array();

        $response['questions'] = SurveyQuestions::all();
        $response['results'] = SurveyResults::all();

        return view('pages.admin.surveyResults')->with($response);
    }

    public function create(){
        $response = array();

        if(Auth::check()){
            if(Auth::user()->survey == 1){
                $response['messageType'] = "warning";
                $response['messageHeading'] = "Not available!";
                $response['message'] = "You have already completed the survey and you
can't complete the survey more than once.";

                return view('layouts.app')->with($response);
            }
            $response['survey'] = SurveyAnswers::all()->groupBy('idQuestion');

            return view('pages.survey')->with($response);
        }
        else{
            $response['messageType'] = "warning";
            $response['messageHeading'] = "Not available!";
            $response['message'] = "You must be logged in to take this survey.";

            return view('layouts.app')->with($response);
        }
    }

    public function submit(Request $request)
    {
        $request->validate([
            'question1' => 'required|exists:surveyanswers,idAnswer',
            'question2' => 'required|exists:surveyanswers,idAnswer'
        ]);

        $idAnswer1 = $request->input('question1');
        $idAnswer2 = $request->input('question2');
        $comment = $request->input('comment');

        $surveyResult = new SurveyResults();

        $surveyResult->idUser = Auth::user()->idUser;
        $surveyResult->idAnswer1 = $idAnswer1;
        $surveyResult->idAnswer2 = $idAnswer2;
        $surveyResult->comment = $comment;

        $surveyResult->save();

        $user = Auth::user();

        $user->survey=1;

        $user->save();

        http_response_code(200);

        $response = array();
        $response['messageType']="success";
        $response['messageHeading']="Well Done!";
        $response['message']="You have successfully completed the survey.";

        return view('include.message')->with($response);
    }
}
```

OfferController

```
<?php

namespace App\Http\Controllers;

use App\Countries;
use App\Locations;
use App\Offers;
use App\Pictures;
use App\Seasons;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;

class OfferController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $response = array();

        $response['countries'] = Countries::all();
        $response['seasons']  = Seasons::all();
        $response['offers'] = Offers::paginate(3);

        return view('pages.offers')->with($response);
    }

    public function indexTable(){
        $response = array();

        $response['offers']=Offers::all();

        return view('pages.admin.offers')->with($response);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $response = array();
        $response['countries'] = Countries::all();
        $response['seasons']  = Seasons::all();

        return view('pages.admin.insert.offer')->with($response);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'country' => 'required|exists:countries,idCountry',
            'location' => 'required|exists:locations,idLocation',
            'season' => 'required|exists:seasons,idSeason',
            'price' => 'required|numeric',
            'description' => 'required',
            'picture' => 'required|image|dimensions:ratio=16/9',
            'pictureAlt' => 'required'
        ]);

        $country = $request->input('country');
        $location = $request->input('location');
        $season = $request->input('season');
        $price = $request->input('price');
        $description = $request->input('description');
        $picture = $request->file('picture');
        $pictureAlt = $request->input('pictureAlt');

        $image = New Pictures();

        $path="/images/".time().$picture->getClientOriginalName();
        $picture->move(public_path()."/images/",time().$picture-
>getClientOriginalName());
        $image->src = $path;
        $image->alt = $pictureAlt;

        $image->save();

        $offer = new Offers();

        $offer->idCountry = $country;
        $offer->idLocation = $location;
        $offer->idSeason = $season;
        $offer->price = $price;
        $offer->description = $description;
        $offer->idPicture = $image->idPicture;

        $offer->save();

        return redirect('/admin');
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $offer = Offers::find($id);
        if($offer == null) {
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following offer with id: ".$id." doesn't exist.";
            return view('layouts.app')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }
        return view('pages.offer')->with(['offer'=>$offer]);
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $offer = Offers::find($id);

        if($offer == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following offer with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }

        $response = array();

        $response['offer'] = $offer;
        $response['countries'] = Countries::all();
        $response['locations'] = Locations::all()->where('idCountry','=',$offer-
>idCountry);
        $response['seasons'] = Seasons::all();

        return view('pages.admin.edit.offer')->with($response);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $request->validate([
            'country' => 'required|exists:countries,idCountry',
            'location' => 'required|exists:locations,idLocation',
            'season' => 'required|exists:seasons,idSeason',
            'price' => 'required|numeric',
            'description' => 'required',
            'picture' => 'sometimes|image|dimensions:ratio=16/9',
            'pictureAlt' => 'required'
        ]);

        $country = $request->input('country');
        $location = $request->input('location');
        $season = $request->input('season');
        $price = $request->input('price');
        $description = $request->input('description');
        $pictureAlt = $request->input('pictureAlt');

        $offer = Offers::find($id);

        $offer->idCountry = $country;
        $offer->idLocation = $location;
        $offer->idSeason = $season;
        $offer->price = $price;
        $offer->description = $description;

        $offer->save();
        $image = Pictures::find($offer->idPicture);

        $image->alt = $pictureAlt;
        if($request->hasFile('picture')){

            $picture = $request->file('picture');
            $picture->move(public_path()."/images/",time().$picture-
>getClientOriginalName());

            File::delete(public_path().$image->src);

            $path="/images/".time().$picture->getClientOriginalName();
            $image->src = $path;
        }
        $image->save();

        return redirect('/admin');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $offer = Offers::all()->find($id);
        if($offer == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following offer with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }
        else{
            $offer->delete();

            return redirect('/admin');
        }
    }

    public function search(Request $request){
        $response = array();

        $response['countries'] = Countries::all();
        $response['seasons'] = Seasons::all();

        $idCountry = $request->input('country');
        $idLocation = $request->input('location');
        $idSeason = $request->input('season');
        $minPrice = $request->input('minPrice');
        $maxPrice = $request->input('maxPrice');
        $sort = $request->input('sort');

        $offers = Offers::all();

        if(is_numeric($idCountry)) $offers = $offers-
>where('idCountry','=',$idCountry);
        if(is_numeric($idLocation)) $offers = $offers-
>where('idLocation','=',$idLocation);
```

```
        if(is_numeric($idSeason)) $offers = $offers->where('idSeason','=',$idSeason);
        if(is_numeric($minPrice)) $offers = $offers->where('price','>=',$minPrice);
        if(is_numeric($maxPrice)) $offers = $offers->where('price','<=',$maxPrice);
        switch ($sort){
            case "locationNameASC": $offers = $offers-
>sortBy('location.locationName'); break;
            case "locationNameDESC": $offers = $offers-
>sortByDesc('location.locationName'); break;
            case "priceASC": $offers = $offers->sortBy('price'); break;
            case "priceDESC": $offers = $offers->sortByDesc('price'); break;
        }

        if($offers->count() == 0){
            $response['messageType']='warning';
            $response['messageHeading']='No Results!';
            $response['message']='There are no offers with these search parameters.';
        }
        $response['offers'] = $offers;
        $response['noPagination'] = 1;

        return view('pages.offers')->with($response);
    }
}
```

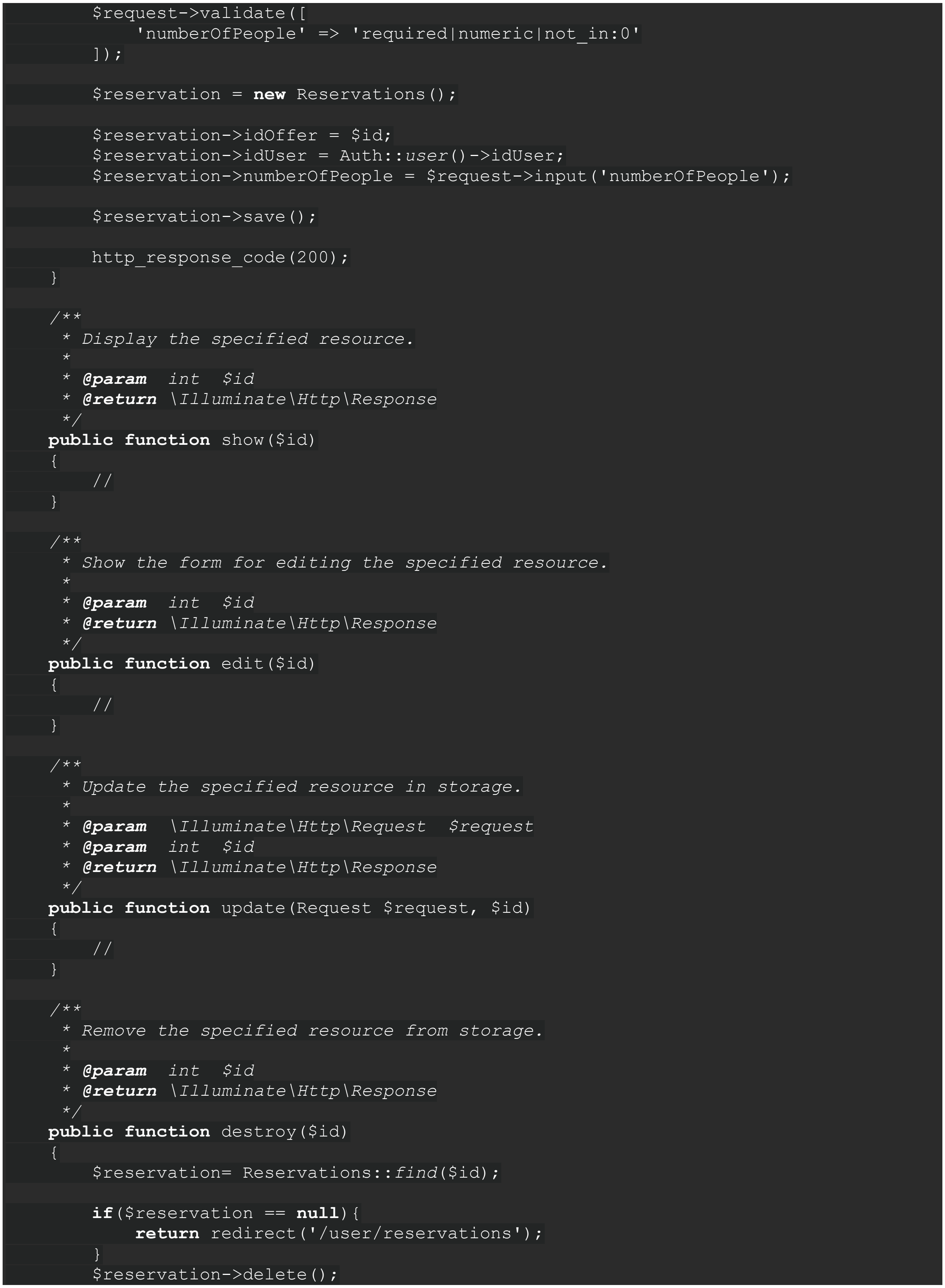
SlideController

```
<?php

namespace App\Http\Controllers;

use App\Pictures;
use App\Slides;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;
use PhpParser\Node\Expr\AssignOp\ShiftLeft;

class SlideController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $response= array();

        $response['slides'] = Slides::all();

        return view('pages.admin.slides')->with($response);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('pages.admin.insert.slide');
    }

    /**
```

```
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'caption' => 'required',
            'subCaption' => 'required',
            'picture' => 'required|image|dimensions:ratio=16/9',
            'pictureAlt' => 'required'
        ]);

        $caption = $request->input('caption');
        $subCaption = $request->input('subCaption');
        $picture = $request->file('picture');
        $pictureAlt = $request->input('pictureAlt');

        $image = New Pictures();

        $path="/images/".time().$picture->getClientOriginalName();
        $picture->move(public_path()."/images/",time().$picture-
>getClientOriginalName());
        $image->src = $path;
        $image->alt = $pictureAlt;

        $image->save();

        $slide = new Slides();

        $slide->caption = $caption;
        $slide->subCaption = $subCaption;
        $slide->idPicture = $image->idPicture;

        $slide->save();

        return redirect('/admin/slides');
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $slide = Slides::find($id);

        if($slide == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following slide with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }

        $response = array();

        $response['slide'] = $slide;

        return view('pages.admin.edit.slide')->with($response);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $request->validate([
            'caption' => 'required',
            'subCaption' => 'required',
            'picture' => 'sometimes|image|dimensions:ratio=16/9',
            'pictureAlt' => 'required'
        ]);

        $slide = Slides::find($id);

        $slide->caption = $request->input('caption');
        $slide->subCaption = $request->input('subCaption');

        $slide->save();

        $image = Pictures::find($slide->idPicture);

        $image->alt = $request->input('pictureAlt');
        if($request->hasFile('picture')){

            $picture = $request->file('picture');
            $picture->move(public_path()."/images/",time().$picture-
>getClientOriginalName());

            File::delete(public_path().$image->src);

            $path="/images/".time().$picture->getClientOriginalName();
            $image->src = $path;
        }
        $image->save();

        return redirect('/admin/slides');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $slide = Slides::all()->find($id);
        if($slide == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following slide with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }
        else{
            $slide->delete();

            return redirect('/admin/slides');
        }
    }
}
```

ReservationController

```
<?php

namespace App\Http\Controllers;

use App\Reservations;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class ReservationController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $response = array();

        $response['reservations'] = Reservations::all()-
>where('idUser','=',Auth::user()->idUser);

        return view('pages.reservations')->with($response);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request, $id)
    {
```



```
        $request->validate([
            'numberOfPeople' => 'required|numeric|not_in:0'
        ]);

        $reservation = new Reservations();

        $reservation->idOffer = $id;
        $reservation->idUser = Auth::user()->idUser;
        $reservation->numberOfPeople = $request->input('numberOfPeople');

        $reservation->save();

        http_response_code(200);
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $reservation= Reservations::find($id);

        if($reservation == null){
            return redirect('/user/reservations');
        }
        $reservation->delete();
        return redirect('/user/reservations');
    }
}
```

CountryController

```
<?php

namespace App\Http\Controllers;

use App\Countries;
use Illuminate\Http\Request;

class CountryController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $response = array();

        $response['countries'] = Countries::all();

        return view('pages.admin.countries')->with($response);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('pages.admin.insert.country');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'country' => 'required|unique:countries,countryName'
        ]);

        $country = new Countries();

        $country->countryName = $request->input('country');

        $country->save();

        return redirect('/admin/countries');
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $country = Countries::find($id);

        if($country == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following location with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }

        $response = array();

        $response['country'] = $country;

        return view('pages.admin.edit.country')->with($response);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $request->validate([
            'country' => 'required|unique:countries,countryName'
        ]);

        $country = Countries::find($id);

        $country->countryName = $request->input('country');

        $country->save();

        return redirect('/admin/countries');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $country = Countries::all()->find($id);
        if($country == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following country with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }
        else{
            $country->delete();

            return redirect('/admin/countries');
        }
    }
}
```

LocationController

```
<?php

namespace App\Http\Controllers;

use App\Countries;
use App\Locations;
use Illuminate\Http\Request;

class LocationController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $response = array();

        $response['locations'] = Locations::all();

        return view('pages.admin.locations')->with($response);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $response = array();

        $response['countries'] = Countries::all();

        return view('pages.admin.insert.location')->with($response);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'country' => 'required|exists:countries,idCountry',
            'location' => 'required'
        ]);

        $location = new Locations();

        $location->idCountry = $request->input('country');
        $location->locationName = $request->input('location');

        $location->save();

        return redirect('/admin/locations');
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $location = Locations::find($id);

        if($location == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following location with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }

        $response = array();

        $response['location'] = $location;
        $response['countries'] = Countries::all();

        return view('pages.admin.edit.location')->with($response);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $request->validate([
            'country' => 'required|exists:countries,idCountry',
            'location' => 'required'
        ]);

        $location = Locations::find($id);

        $location->idCountry = $request->input('country');
        $location->locationName = $request->input('location');

        $location->save();

        return redirect('/admin/locations');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $location = Locations::all()->find($id);
        if($location == null){
            $messageType="danger";
            $messageHeading = "Error!";
            $message = "The following location with id: ".$id." doesn't exist.";
            return view('layouts.admin')-
>with(['messageType'=>$messageType,'messageHeading'=>$messageHeading,'message'=>$messa
ge]);
        }
        else{
            $location->delete();

            return redirect('/admin/locations');
        }
    }

    public function filter(Request $request){
        $locations = Locations::all()->where('idCountry',$request-
>input('idCountry'));
        return view('include.locations')->with('locations',$locations);
    }
}
```

ContactController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ContactController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return view('pages.contact');
    }

    public function submit(Request $request)
    {
        $nameRegex = "/^[a-zA-
ZàáâäãåąčćęèéêëėįìíîïłńòóôöõøùúûüųūÿýżźñçčšžÀÁÂÄÃÅĄĆČĖĘÈÉÊËÌÍÎÏĮŁŃÒÓÔÖÕØÙÚÛÜŲŪŸÝŻŹÑßÇŒ
ÆČŠŽ∂ð ,.'-]+$/u";
        $emailRegex = "/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0-9\-]+\.)+[a-
z]{2,6}$/ix";

        $request->validate([
            'fullName' => 'required|regex:'.$nameRegex,
            'eMail' => 'required|regex:'.$emailRegex,
            'title' => 'required',
            'message' => 'required',
        ]);

        $fullName = $request->input('fullName');
        $eMail = $request->input('eMail');
        $title = $request->input('title');
        $messageToAdmin = $request->input('message');

        $to = "joca296burner@gmail.com";
        $title = "Fast Travel contact page | ".$title;
        $messageToAdmin = "From: ".$fullName."\r\nE-mail address:
".$eMail."\r\nMessage:\r\n".$messageToAdmin;
        mail($to,$title,$messageToAdmin);

        $return = array();
        $return['messageType'] = 'success';
        $return['messageHeading'] = 'Success';
        $return['message'] = 'You have sent an e-mail to the site administrator.';

        return view('pages.contact')->with($return);
    }

}
```

PageController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller
{
    public function about(){
        return view('pages.about');
    }
}
```