

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I  
KOMUNIKACIONE TEHNOLOGIJE



## WEB PROGRAMIRANJE PHP2

## DOKUMENTACIJA PROJEKTA

Student  
Bogdan Matorkić 30/13

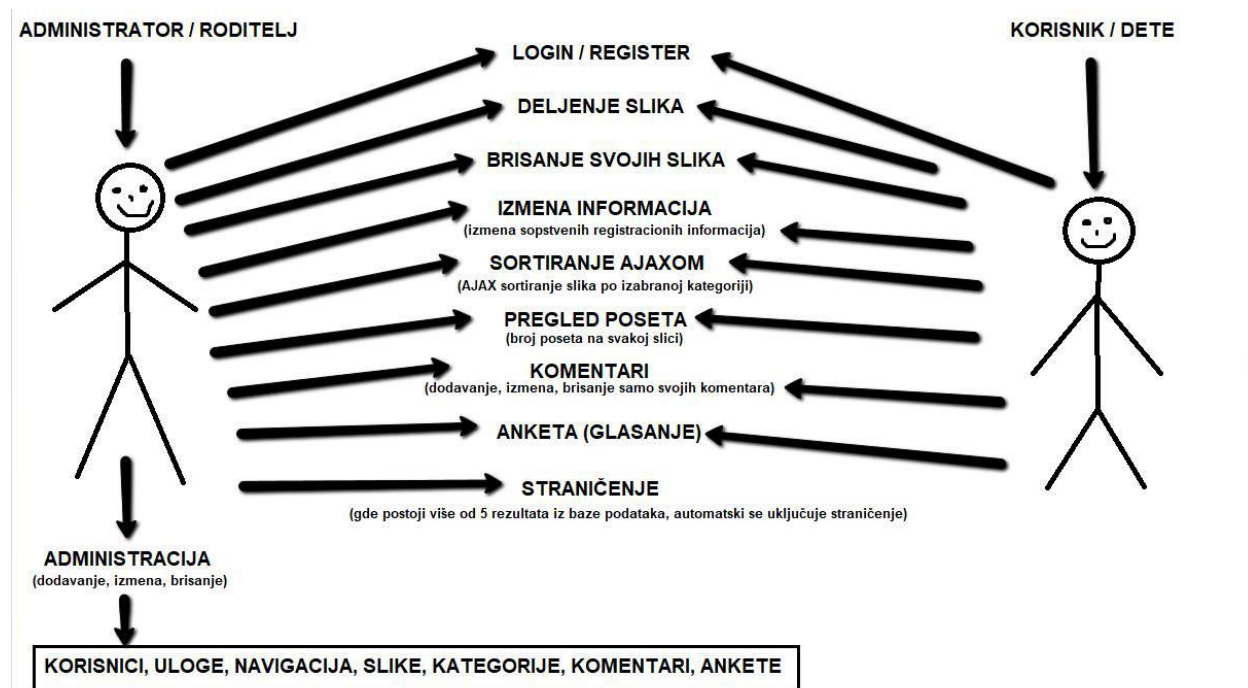
Beograd, Mart 2018

## SADRŽAJ

WEB PROGRAMIRANJE PHP2.....	1
DOKUMENTACIJA PROJEKTA .....	1
1. Opis funkcionalnosti .....	4
2. Skica struktura stranice .....	6
2.1 Deo za korisnika .....	6
2.2 Deo za administratora .....	6
3. Dijagram baze podataka.....	7
3. MVC ORGANIZACIJA .....	8
KONTROLERI .....	8
MODELI .....	10
VIEW .....	12
4. javaScript kod .....	13
public/js/my.js .....	13
public/js/ajax.js .....	15
5. PHP kod .....	21
AdminController .....	21
BookingController .....	24
CategoryController .....	30
CommentController .....	32
FrontendController.....	36
NavigationController .....	39
PictureController.....	42
PollController.....	48
RoleController.....	53
RoleController.....	55
Middleware/AdminMiddleware.....	63
Middleware/UserMiddleware .....	63
Models/Category.php.....	65
Models/Comment.php.....	67
Models/Navigation.php.....	69
Models/Picture.php .....	71

Models/Poll.php .....	75
Models/Role.php .....	80
Models/User.php .....	82
layout/template.blade.php .....	85
components/navigation.blade.php .....	86
components/footer.blade.php .....	87
pages/admin.blade.php .....	87
pages/gallery.blade.php .....	88
pages/home.blade.php .....	88
pages/information.blade.php .....	89
pages/picture.blade.php .....	90
pages/share.blade.php .....	93
admin/adminCategory.blade.php .....	95
admin/adminNavigation.blade.php .....	97
admin/adminPicture.blade.php .....	99
admin/adminPolls.blade.php .....	102
admin/adminRoles.blade.php .....	104
admin/adminUser.blade.php .....	106
ajax/ajaxGallery.blade.php .....	108
routes/web.php .....	109

## 1. Opis funkcionalnosti



<b>Korišćeni jezici</b>	<ul style="list-style-type: none"> <li>• HTML 4 i HTML 5</li> <li>• CSS</li> <li>• JavaScript</li> <li>• jQuery</li> <li>• PHP</li> </ul>
<b>Korišćeni HTML šabloni (framework-ci)</b>	<ul style="list-style-type: none"> <li>• <a href="https://w3layouts.com/game-robo-a-games-category-bootstrap-responsive-web-template/">https://w3layouts.com/game-robo-a-games-category-bootstrap-responsive-web-template/</a></li> </ul>
<b>Korišćene biblioteke i njihovi url-ovi odakle ste ih preuzeli</b>	<ul style="list-style-type: none"> <li>• jQuery</li> <li>• Sve biblioteke koje sam koristio su već bile u okviru template-a!</li> </ul>
<b>Korišćene php biblioteke van framework-a</b>	<ul style="list-style-type: none"> <li>• Intervention Image</li> </ul>
<b>Korišćene biblioteke iz framework-a</b>	<ul style="list-style-type: none"> <li>• Laravel</li> </ul>
<b>Koje funkcionalnosti su realizovane putem AJAX-a</b>	<ul style="list-style-type: none"> <li>• <b>Straničenje</b> koje se automatski prikazuje kod svakog tabelarnog prikaza u admin panelu <b>gde ima više od 5 rezultata</b></li> <li>• Sortiranje slika po kategoriji</li> <li>• Prikaz ankete</li> <li>• Prikaz odgovora ankete</li> <li>• Glasanje u anketi</li> </ul>
<b>Navesti lokacije gde je napisan AJAX kod (fajlovi ili delovi html stranice)</b>	<ul style="list-style-type: none"> <li>• public/js/ajax.js</li> </ul>
<b>Navesti lokacije gde je napisan sopstveni</b>	

<b>javaScript kod (fajlovi ili delovi html stranice)</b>	<ul style="list-style-type: none"> <li>• public/js/my.js</li> <li>• public/js/ajax.js</li> </ul>
<b>URL sajta/aplikacije</b>	<a href="https://snapshotshare.000webhostapp.com/home">https://snapshotshare.000webhostapp.com/home</a>
<b>Pristupni parametri</b>	<ul style="list-style-type: none"> <li>• Poslati na e-mail</li> </ul>

## 2. Skica struktura stranice

### 2.1 Deo za korisnika

**Share something and contribute to our community!**

Keep in mind that there are few rules:

1. Name must be unique.
2. Picture must be in .jpg or .jpeg format.
3. Be original, do not share pictures from other users.
4. Enjoy!

Name:  Picture:  No file selected.

Category:

**Your pictures**

**Change profile**

E-Mail:  Username:

Password:

**NAVIGACIJA**

**FUTER**

Home Gallery Share Admin Panel Information Logout

Copyright © Bogdan Matokić 2018, Snapshot

### 2.2 Deo za administratora

**Admin Panel**

**NAVIGACIJA**

**NAVIGACIJA ADMIN PANELA**

**MANAGE PICTURES**

ID	Picture	Name	Category	User	Shared	Updated	Edit	Delete
1	PREVIEW	Asian House	Places	pera	10.01.2018	11.03.2018	<input type="button" value="edit"/>	<input type="button" value="delete"/>
2	PREVIEW	Asian Road	Places	ana	05.12.2017	NEVER	<input type="button" value="edit"/>	<input type="button" value="delete"/>
3	PREVIEW	Bag	Objects	mika	15.11.2017	11.03.2018	<input type="button" value="edit"/>	<input type="button" value="delete"/>
4	PREVIEW	Bike	Objects	mika	07.02.2018	NEVER	<input type="button" value="edit"/>	<input type="button" value="delete"/>
5	PREVIEW	Car	Objects	user	22.12.2017	NEVER	<input type="button" value="edit"/>	<input type="button" value="delete"/>

**ADD PICTURE**

Name:  Picture:  No file selected.

Category:  User:

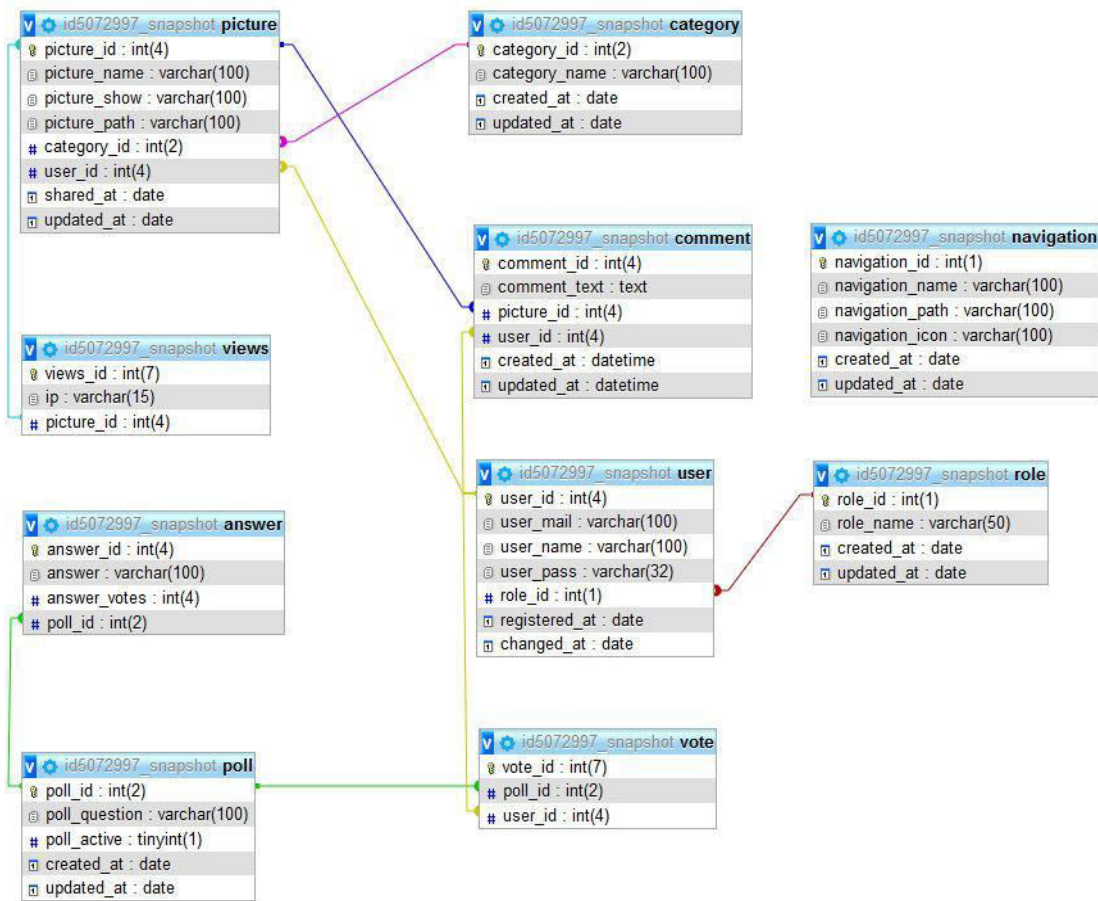
**CENTRALNI SADRŽAJ ADMIN PANELA**

**FUTER**

Home Gallery Share Admin Panel Information Logout

Copyright © Bogdan Matokić 2018, Snapshot

### 3. Dijagram baze podataka



### 3. MVC ORGANIZACIJA

#### KONTROLERI

<b>AdminController</b>
__construct()
showUsers(\$id?)
showPictures(\$id?)
showCategories(\$id?)
showNavigation(\$id?)
showRoles(\$id?)
showPolls(\$id?)

<b>CategoryController</b>
insertCategory(\$request)
updateCategory(\$request, \$id)
deleteCategory(\$id)

<b>CommentController</b>
insertComment(\$request, \$id)
updateComment(\$id, \$cId, \$request)
deleteComment(\$request, \$id, \$cId)

<b>FrontendController</b>
__construct()
getHome(\$request)
getGallery()
showCategorized(\$cat)
getShare()
getAdmin()
getInformation()
getPicture(\$request, \$id, \$cId?)

<b>NavigationController</b>
insertNavigation(\$request)
updateNavigation(\$request, \$id)
deleteNavigation(\$id)

<b>PictureController</b>
insertPicture(\$request)
updatePicture(\$request, \$id)
deletePicture(\$id)
sharePicture(\$request)
deleteMyPicture(\$request, \$id)
store(\$request)
commentInsert(\$request)
commentDestroy(\$request, \$id)

<b>PollController</b>
showPoll()
insertPoll(\$request)
updatePoll(\$request, \$id)
deletePoll(\$id)
insertVote(\$request)
activatePoll(\$id)



<b>RoleController</b>
insertRole(\$request)
updateRole(\$request, \$id)
deleteRole(\$id)

<b>RolesController</b>
login(\$request)
logout(\$request)
register(\$request)
insertUser(\$request)
updateUser(\$request, \$id)
deleteUser(\$id)
changeUser(\$request, \$id)

## MODEL

Category
show()
get()
insert()
update()
delete()

Comment
getPost(\$id)
get()
insert()
update()
deleteAdminComment()
deleteComment()

Navigation
getAll()
get()
getAuth()
getNoAuth()
insert()
update()
delete()

Picture
getAll()
getLimited()
getCategorized(\$cat)
getUser(\$id)
getSingle(\$id)
get()
insert()
update()
delete()
deleteMyPic()
getViews(\$id, \$ip)

Poll
getAll()
getOne()
get()
findUser()
insertPoll()
updatePoll()
deletePoll()
insertFirstAnswer()
insertSecondAnswer()
updateFirstAnswer()
updateSecondAnswer()
deleteAnswer()
incAnswer()
insertVote()
deleteVote()
activate()
deactivate()

<b>Role</b>
getAll()
get()
insert()
update()
delete()

<b>User</b>
getAll()
getUser()
getProfile(\$id)
get()
insert()
update()
delete()
change()

## VIEW

layout
template.blade.php

components
navigation.blade.php
footer.blade.php

pages
admin.blade.php
gallery.blade.php
home.blade.php
information.blade.php
picture.blade.php
share.blade.php

admin
adminCategory.blade.php
adminNavigation.blade.php
adminPicture.blade.php
adminPolls.blade.php
adminRoles.blade.php
adminUser.blade.php

ajax
ajaxGallery.blade.php

## 4. JavaScript kod

### public/js/my.js

```
(function($) {  
  
    'use strict';  
  
    $(document).ready(function() {  
        close();  
        show();  
        active();  
        slide();  
    });  
  
    function close() {  
        var closeIcon = $(' .close');  
  
        closeIcon.click(function() {  
            $(this).parents(' .error').fadeOut();  
        });  
  
        closeIcon.click(function() {  
            $(this).parents(' .success').fadeOut();  
        });  
    }  
  
    function show() {  
        var show = $(' .show');  
  
        show.click(function() {  
            $(this).children().fadeToggle();  
        });  
    }  
  
    function active() {  
        var button = $(' .button');  
  
        button.click(function() {  
            $(this).addClass('active');  
            $(this).parent().siblings().children().removeClass('active'); });  
    }  
})
```

```

function slide() {
    $('.arrow-down').click(function() {
        $(this).parent().animate({ top: '0px'}).delay(1000);
        $(this).parent().animate({ top: '-67px'});
    });
}

})(jQuery);

function checkMail()
{
    var mail = document.getElementById('emailRegister').value;
    var regMail = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

    if(!regMail.test(mail))
    {
        document.getElementById('emailRegister').style.border = '2px solid red';
        document.getElementById('emailRegister').style.color = 'red';
    }
    else
    {
        document.getElementById('emailRegister').style.border = '2px solid green';
        document.getElementById('emailRegister').style.color = 'black';
    }
}

function checkUsername()
{
    var username = document.getElementById('usernameRegister').value;
    var regUsername = /^[a-zA-Z0-9]{3,15}$/;

    if(!regUsername.test(username))
    {
        document.getElementById('usernameRegister').style.border = '2px solid red';
        document.getElementById('usernameRegister').style.color = 'red'; }

    else
    {
        document.getElementById('usernameRegister').style.border = '2px solid green';
        document.getElementById('usernameRegister').style.color = 'black'; }

}

```

```

function checkPassword()
{
    var password = document.getElementById('passwordRegister').value;
    var regPassword = /^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{6,}$/;

    if(!regPassword.test(password))
    {
        document.getElementById('passwordRegister').style.border = '2px solid red';
        document.getElementById('passwordRegister').style.color = 'red';
    }
    else
    {
        document.getElementById('passwordRegister').style.border = '2px solid green';
        document.getElementById('passwordRegister').style.color = 'black';
    }
}

function checkPasswordConfirm()
{
    var password = document.getElementById('passwordRegister').value;
    var password2 = document.getElementById('passwordRegister_confirmation').value;

    if(!password.match(password2))
    {
        document.getElementById('passwordRegister_confirmation').style.border = '2px solid red';
        document.getElementById('passwordRegister_confirmation').style.color = 'red';
    }
    else
    {
        document.getElementById('passwordRegister_confirmation').style.border = '2px solid green';
        document.getElementById('passwordRegister_confirmation').style.color = 'black';
    }
}

```

## public/js/ajax.js

```

$(document).ready(function(){
    $('a[id^="cat"]').click(function(e){
        e.preventDefault();
        var c = $(this).attr('id').slice(3);
        $.ajax({

```

```

        type: "GET",
        url: "http://localhost/Sajt/public/gallery/" + c,
        success: function(data) {
            $('#content').html(data);
        }
    });
});

$(".wrapper-paging").show();
TABLE.paginate('.table', 5);
});

var TABLE = {};

TABLE.paginate = function(table, pageLength) {
    var $table = $(table);
    var $rows = $table.find('tbody > tr');
    var numPages = Math.ceil($rows.length / pageLength) - 1;
    var current = 0;

    var $nav = $table.parents('.table-wrapper').find('.wrapper-paging ul');
    var $back = $nav.find('li:first-child a'); var $next = $nav.find('li:last-child a');

    $nav.find('a.paging-this strong').text(current + 1);
    $nav.find('a.paging-this span').text(numPages + 1);
    $back.addClass('paging-disabled').click(function() {
        pagination('<');
    });

    $next.click(function() {
        pagination('>');
    });

    $rows.hide().slice(0, pageLength).show();

    pagination = function(direction) {
        reveal = function(current) {
            $back.removeClass('paging-disabled');
            $next.removeClass('paging-disabled');
            $rows.hide().slice(current * pageLength, current * pageLength +
                pageLength).show(); $nav.find('a.paging-this strong').text(current + 1);
        };
    };
};

```



```

        if(direction === '<'){
            if(current > 1){
                reveal(current -= 1);
            }
            else if(current === 1){
                reveal (current -= 1);
                $back.addClass('paging-disabled');
            }
        }
        else{
            if(current < numPages - 1){
                reveal(current += 1);
            }
            else if(current === numPages - 1){
                reveal(current += 1);
                $next.addClass('paging-disabled');
            }
        }
    };
};

$(document).ready(function(){
    var exist;

    element = document.getElementById('exist');
    console.log(element);

    if(element != null)
    {
        exist = element.value;
    }
    else
    {
        exist = null;
    }

    $.ajax({
        type: 'GET',
        url: baseUrl + 'home/showPoll',
        success: function(data, xhr)
        {
            console.log(data);

```

```

        console.log(xhr);

        if(exist == 0)
        {
            showPoll(data);
        }
        else
        {
            $('#poll').html(showAnswers());
            $('#feedback').html('You have already voted!');
        }
    },
    error: function(xhr, status, error)
    {
        console.log(xhr);
        console.log(status);
        console.log(error);
    }
});
});

function showAnswers()
{
    $.ajax({
        type: 'GET',
        url: baseUrl + 'home/showPoll',
        success : function(data)
        {
            var html = `<h2>`+data[0].poll_question+`</h2>`;
            $.each(data, function(key, value){
                html+=`<div>
                    <h3>`+value.answer+`:
                    `+value.answer_votes+`</h3> </div>`;
            });
            $('#poll').html(html);
        },
        error: function(errors)
        {
            console.log(errors);
        }
    });
}

```

```

function showPoll(data){
    var html = "";
    html+=`<form action=" method=">`;
    html+=`<h3>`+data[0].poll_question+`</h3><input type='hidden'
value=`+data[0].poll_id+`' id='idPoll'></input>`;
    $.each(data, function(key, value){
        html+=`<div>
            <input type="radio" name="answers" value="`+value.answer_id+`">
            <label>`+value.answer+`</label>
        </div>`;
    });
    html+=`<div>
        <input type="button" class="button" value="VOTE"
        onclick="vote();" /> </div>`;
    html+=`</form>`;
    $('#poll').html(html);
}

```

```

function vote()
{
    var idUser = document.getElementById('idUser').value;
    var idPoll = document.getElementById('idPoll').value;
    var idAnswer = $('input[name=answers]:checked').val();
    if(document.getElementsByName('answers')[0].checked ||
document.getElementsByName('answers')[1].checked)
    {
        $.ajax({
            type: 'POST',
            url: baseUrl + 'home/insertVote',
            data:
            {
                _token: token,
                user_id: idUser,
                poll_id: idPoll,
                answer_id: idAnswer
            },
            success: function(data, xhr)
            {
                console.log(data);

                $('#feedback').html('You voted successfully!');
                $('#poll').html(showAnswers());
            },

```

```

        error: function(xhr, status, error)
        {
            console.log(error);
            $('#poll').html(showAnswers());
        }
    });

    console.log(baseUrl + 'home/insertVote');
    console.log(token);
    console.log(idPoll + ' ' + idUser + ' ' + idAnswer);
}
else
{
    $('#feedback').html('<b>You must choose something!</b>');
}
}

```

## 5. PHP kod

### AdminController

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Role;
use App\Models\Picture;
use App\Models\Category;
use App\Models\Navigation;
use App\Models\Poll;

class AdminController extends Controller
{
    private $data = [];

    public function __construct()
    {
        $navigation = new Navigation();
        $this->data['navigations'] = $navigation->getAll();
    }

    public function showUsers($id = null)
    {
        try
        {
            $user = new User();
            $role = new Role();
            $user->user_id = $id;
            $this->data['users'] = $user->getAll();
            $this->data['roles'] = $role->getAll();
            $this->data['selectedUser'] = $user->get();
            return view('admin.adminUser', $this->data);
        }
        catch(Exception $ex)
        {
            \Log::error($ex->getMessage());
        }
    }
}
```

```

public function showPictures($id = null)
{
    try
    {
        $picture = new Picture();
        $category = new Category();
        $user = new User();
        $picture->picture_id = $id;
        $this->data['pictures'] = $picture->getAll();
        $this->data['categories'] = $category->getAll();
        $this->data['users'] = $user->getAll();
        $this->data['selectedPicture'] = $picture->get();
        return view('admin.adminPicture', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function showCategories($id = null)
{
    try
    {
        $category = new Category();
        $category->category_id = $id;
        $this->data['categories'] = $category->getAll();
        $this->data['selectedCategory'] = $category->get();
        return view('admin.adminCategory', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function showNavigation($id = null)
{
    try
    {
        $navigation = new Navigation();
        $navigation->navigation_id = $id;
        $this->data['navigations'] = $navigation->getAll();
    }
}

```

```

        $this->data['selectedNavigation'] = $navigation->get();
        return view('admin.adminNavigation', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

public function showRoles($id = null)
{
    try
    {
        $role = new Role();
        $role->role_id = $id;
        $this->data['roles'] = $role->getAll();
        $this->data['selectedRole'] = $role->get();
        return view('admin.adminRoles', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

public function showPolls($id = null)
{
    try
    {
        $poll = new Poll();
        $poll->poll_id = $id;
        $this->data['polls'] = $poll->getAll();

        if(!empty($id))
        {
            $this->data['selectedPoll'] = $poll->getOne();
        }

        return view('admin.adminPolls', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

    }
}
}

```

## BookingController

```

<?php

namespace App\Http\Controllers;

use App\Models\Booking;
use App\Models\Datum;
use App\Models\Destination;
use App\Models\Navigation;
use App\Models\Users;
use Illuminate\Http\Request;

class BookingController extends Controller
{
    private $data = [];

    public function __construct() {
        $nav = new Navigation();
        $des = new Destination();
        $user = new Users();
        $datum = new Datum();
        $this->data['admin'] = $nav->getAll();
        $this->data['user'] = $nav->getUser();
        $this->data['noUser'] = $nav->getNoUser();
        $this->data['getDes'] = $des->getAll();
        $this->data['getUser'] = $user->showAll();
        $this->data['getDatum'] = $datum->getDatum();
    }

    public function insert(Request $request)
    {
        $request->validate([
            'mojSelect' => 'not_in:0',
        ], [
            'mojSelect.not_in' => '*Choose date!',
        ]);
        $idDestination = $request->get('idDestination');
    }
}

```



```

$idUser = $request->get('idUser');
$idDate = $request->get('mojSelect');
$inserted_at = date("Y-m-d H:i:s");

try {
    $book = new Booking();
    $book->id_destination = $idDestination;
    $book->id_user = $idUser;
    $book->id_date = $idDate;
    $book->inserted_at = $inserted_at;

    $rez = $book->save();

    if($rez == 1)
    {
        return redirect('/destination/'.$idDestination)->with('success','You have
successfully booked your trip!');
    }
    else
    {
        return redirect('/destination/'.$idDestination)->with('error','There\' been an error!');
    }

} catch (\Exception $ex) {
    \Log::error('Error while enetering booking in db! '.$ex->getMessage());
}

}

public function show(Request $request)
{
    try {
        if(session()->has('user'))
        {
            $id = $request->session()->get('user')[0]->id;
            $book = new Booking();
            $this->data['book'] = $book->getRes($id);
            $this->data['price'] = $book->getPrice($id);
            return view('pages.booking', $this->data);
        }
        else
        {
            return redirect('401');
        }
    }
}

```

```

    }
} catch (\Exception $ex) {
    \Log::error('Greska prilikom prikaza rezervacije! '.$ex->getMessage());
}
}

public function cancelRes(Request $request, $id)
{
    try {
        if(session()->has('user'))
        {
            $book = new Booking();
            $book->id_user = $request->session()->get('user')[0]-
            >id; $book->id_book = $id;
            $rez = $book->deleteRes();

            if($rez == 1){
                return redirect('/mybooking')->with('success','You have successfully canceled
your reservation!');
            }
            else {
                return redirect('404');
            }
        }
        else
        {
            return redirect('401');
        }
    } catch (\Exception $ex) {
        \Log::error('Greska prilikom brisanja rezervacije! '.$ex->getMessage());
    }
}

public function showAll($id = null)
{
    try {
        $book = new Booking();
        $this->data['book'] = $book->getAll();

        if(!empty($id))
        {
            $book->id_book = $id;

```

```

        $this->data['oneRes'] = $book->getOne();
    }
    return view('pages.adminBooking', $this->data); } catch (\Exception $ex) {
        \Log::error('Greska prilikom prikaza rezervacije! '.$ex->getMessage());
    }
}

public function destroy($id)
{
    try {
        $book = new Booking();
        $book->id_book = $id;

        $rez = $book->deleteBook();
        if($rez == 1)
        {
            return redirect('/admin/booking')->with('success', 'Reservation successfully deleted!');
        }
        else
        {
            return redirect('/admin/booking')->with('error', 'Reservation could not be deleted!');
        }
    } catch (\Exception $ex) {
        \Log::error('Greska prilikom brisanja rezervacije! '.$ex->getMessage());
    }
}

public function update($id, Request $request)
{
    $idDes = $request->get('ddlDes');
    $idUser = $request->get('ddlUser');
    $idDate = $request->get('ddlDate');
    $inserted_at = date('Y-m-d H:i:s');

    try {
        $book = new Booking();
        $book->id_book = $id;
        $book->id_destination = $idDes;
        $book->id_user = $idUser;
        $book->id_date = $idDate;
        $book->inserted_at = $inserted_at;
    }
}

```

```

    $rez = $book->updateBook();
    if($rez == 1)
    {
        return redirect('/admin/booking')->with('success', 'Reservation successfully updated!');
    }
    else
    {
        return redirect('/admin/booking')->with('error', 'Reservation could not be updated!');
    }
} catch (\Exception $ex) {
    \Log::error('Neuspesna izmena rezervacije! '.$ex->getMessage());
}
}

public function store(Request $request)
{
    $idDes = $request->get('ddlDes');
    $idUser = $request->get('ddlUser');
    $idDate = $request->get('ddlDate');
    $inserted_at = date('Y-m-d H:i:s');

    try {
        $book = new Booking();
        $book->id_destination = $idDes;
        $book->id_user = $idUser;
        $book->id_date = $idDate;
        $book->inserted_at = $inserted_at;

        $rez = $book->save();
        if($rez == 1)
        {
            return redirect('/admin/booking')->with('success', 'Reservation successfully added!');
        }
        else
        {
            return redirect('/admin/booking')->with('error', 'Reservation not added!');
        }
    } catch (\Exception $ex) {
        \Log::error('Neuspesan unos rezervacije! '.$ex->getMessage());
    }
}
}

```



## CategoryController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Category;

use DateTime;

class CategoryController extends Controller
{
    public function insertCategory(Request $request)
    {
        $rules =
        [
            'categoryNameAdmin' => 'required|unique:category,category_name'
        ];

        $messages =
        [
            'required' => 'Field :attribute is required!',
            'categoryNameAdmin.unique' => 'Picture name must be unique!',
        ];

        $request->validate($rules, $messages);

        try
        {
            $category = new Category();

            $date = new DateTime();
            $date->format('d.m.Y');

            $category->category_name = $request->get('categoryNameAdmin');
            $category->created_at = $date;
            $category->updated_at = null;

            $category->insert();

            return redirect('/admin-panel/categories')->with('success', 'Category successfully added!');
```

```

    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while adding category!');
    }
}

public function updateCategory($id, Request $request)
{
    $rules =
    [
        'categoryNameAdmin' => 'required'
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!',
    ];

    $request->validate($rules, $messages);

    try
    {
        $category = new Category();

        $date = new DateTime();
        $date->format('d.m.Y');

        $category->category_id = $id;
        $category->category_name = $request-
        >get('categoryNameAdmin'); $category->updated_at = $date;

        $category->update();

        return redirect('/admin-panel/categories')->with('success', 'Category
successfully updated!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while updating category!');
    }
}

```

```

    }

    public function deleteCategory($id)
    {
        try
        {
            $category = new Category();

            $category->category_id = $id;

            $category->delete();

            return redirect('/admin-panel/categories')->with('success', 'Category
successfully deleted!');
        }
        catch(\Illuminate\Database\QueryException $ex)
        {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Error while deleting category!');
        }
    }
}

```

## CommentController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Comment;

use DateTime;

class CommentController extends Controller
{
    public function insertComment(Request $request, $id)
    {
        $rules = [
            'comment' => 'required'
        ];
    }
}

```



```

    $messages = [
        'required' => 'You must comment something!'
    ];

    $request->validate($rules, $messages);

    try
    {
        $comment = new Comment();

        if(session()->has('user'))
        {
            $userId = session()->get('user')[0]->user_id;
        }

        $date = new DateTime();
        $date->format('d.m.Y H:i');

        $comment->comment_text = $request-
        >get('comment'); $comment->picture_id = $id;
        $comment->user_id = $userId;
        $comment->created_at = $date;
        $comment->updated_at = null;

        $comment->insert();

        return redirect()->back()->with('success', 'You have successfully commented!');
    }
    catch(Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while adding comment!');
    }
}

public function updateComment($id, $cId, Request $request)
{
    $rules =
    [
        'comment' => 'required'
    ];

```

```

$messages =
[
    'required' => 'Field :attribute is required!',
];

$request->validate($rules, $messages);

try
{
    $comment = new Comment();

    $date = new DateTime();
    $date->format('d.m.Y');

    $comment->comment_text = $request-
    >get('comment'); $comment->comment_id = $cId;
    $comment->updated_at = $date;

    $comment->update();

    return redirect('/pictures/'.$id)->with('success', 'You have successfully updated
your comment!');
}
catch(Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while updating comment!');
}
}

public function deleteComment(Request $request, $id, $cId)
{
    try
    {
        $comment = new Comment();

        $comment->comment_id = $cId;

        $comment->user_id = $request->session()->get('user')[0]->user_id;

        $role = $request->session()->get('user')[0]->role_name;

        if($role == 'administrator')

```

```

    {
        $comment->deleteAdminComment();

        return redirect()->back()->with('success', 'Comment successfully deleted!');
    }
    else
    {
        $result = $comment->deleteComment();

        if($result == 1)
        {
            return redirect()->back()->with('success', 'Comment successfully deleted!');
        }
        else
        {
            return redirect(401);
        }
    }
}

return redirect()->back()->with('success', 'You have successfully deleted your
comment!');
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while deleting comment!');
}
}
}

```

## FrontendController

<?php

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Navigation;
use App\Models\User;
use App\Models\Picture;
use App\Models\Category;
use App\Models\Poll;
use App\Models\Comment;

class FrontEndController extends Controller
{
    private $data = [];

    public function __construct()
    {
        $navigation = new Navigation();
        $poll = new Poll();
        $this->data['navigations'] = $navigation->getAll();
        $this->data['authNavigations'] = $navigation->getAuth();
        $this->data['noAuthNavigations'] = $navigation->getNoAuth();
        $this->data['polls'] = $poll->getAll();
    }

    public function getHome(Request $request)
    {
        if($request->session()->has('user'))
        {
            $idUser = $request->session()->get('user')[0]->user_id;
            $poll = new Poll();
            $poll->user_id = $idUser;
            $result = $poll->findUser();
            if($result)
            {
                $this->data['result'] = $result;
            }
        }
        else
        {
            $this->data['result'] = $result;
        }
    }
}
```

```

    }
}

$picture = new Picture();
$this->data['pictures'] = $picture->getLimited();
return view('pages.home', $this->data);
}

public function getGallery()
{
    try
    {
        $picture = new Picture();
        $category = new Category();
        $this->data['pictures'] = $picture->getAll();
        $this->data['categories'] = $category->getAll();
        return view('pages.gallery', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function showCategorized($cat)
{
    try
    {
        $picture = new Picture();
        $catData = $picture->getCategorized($cat);
        return view('ajax.ajaxGallery', array('cats'=>$catData));
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function getShare()
{
    try
    {
        $picture = new Picture();

```

```

$user = new User();
$category = new Category();

if(session()->has('user'))
{
    $id = session()->get('user')[0]->user_id;
    $this->data['pictures'] = $picture->getByUser($id);
    $this->data['loggedUsers'] = $user->getProfile($id);
    $this->data['categories'] = $category->getAll();
    return view('pages.share', $this->data);
}
}
catch(Exception $ex)
{
    \Log::error($ex->getMessage());
}
}

public function getAdmin()
{
    try
    {
        $user = new User();
        $this->data['users'] = $user->getAll();
        return view('pages.admin', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function getInformation()
{
    try
    {
        return view('pages.information', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

public function getPicture(Request $request, $id, $cId = null)
{
    try
    {
        $user = new User();
        $picture = new Picture();
        $picture->getViews($id, $request->ip());
        $comment = new Comment();
        $comment->comment_id = $cId;
        $this->data['users'] = $user->getAll();
        $this->data['pictures'] = $picture->getAll();
        $this->data['singlePicture'] = $picture->getSingle($id);
        $this->data['comments'] = $comment->getByPost($id);
        $this->data['selectedComment'] = $comment->get();
        return view('pages.picture', $this->data);
    }
    catch(Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

public function download()
{
    $headers = array(
        'Content-Type: application/pdf',
    );
    return response()->download(public_path('documentation.pdf'),
'documentation.pdf', $headers);
}
}

```

## NavigationController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Navigation;

use DateTime;

```

```

class NavigationController extends Controller
{
    public function insertNavigation(Request $request)
    {
        $rules =
        [
            'navigationNameAdmin' => 'required|unique:navigation,navigation_name',
            'navigationPathAdmin' => 'required',
            'navigationIconAdmin' => 'required'
        ];

        $messages =
        [
            'required' => 'Field :attribute is required!',
            'navigationNameAdmin.unique' => 'Navigation name must be unique'
        ];

        $request->validate($rules, $messages);

        try
        {
            $navigation = new Navigation();

            $date = new DateTime();
            $date->format('d.m.Y');

            $navigation->navigation_name = $request->get('navigationNameAdmin');
            $navigation->navigation_path = $request->get('navigationPathAdmin');
            $navigation->navigation_icon = $request->get('navigationIconAdmin');
            $navigation->created_at = $date;
            $navigation->updated_at = null;

            $navigation->insert();

            return redirect('/admin-panel/navigation')->with('success', 'Navigation link
successfully added!');
        }
        catch(Illuminate\Database\QueryException $ex)
        {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Error while adding navigation link!');
        }
    }
}

```



```

}

public function updateNavigation($id, Request $request)
{
    $rules =
    [
        'navigationNameAdmin' => 'required',
        'navigationPathAdmin' => 'required',
        'navigationIconAdmin' => 'required'
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!',
    ];

    $request->validate($rules, $messages);

    try
    {
        $navigation = new Navigation();

        $date = new DateTime();
        $date->format('d.m.Y H:i');

        $navigation->navigation_id = $id;
        $navigation->navigation_name = $request-
        >get('navigationNameAdmin'); $navigation->navigation_path = $request-
        >get('navigationPathAdmin'); $navigation->navigation_icon = $request-
        >get('navigationIconAdmin'); $navigation->updated_at = $date;

        $navigation->update();

        return redirect('/admin-panel/navigation')->with('success', 'Navigation link
        successfully updated!');
    }
    catch(Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while updating navigation link!');
    }
}

```

```

public function deleteNavigation($id)
{
    try
    {
        $navigation = new Navigation();

        $navigation->navigation_id = $id;

        $navigation->delete();

        return redirect('/admin-panel/navigation')->with('success', 'Navigation link
successfully deleted!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while deleting navigation link!');
    }
}
}

```

## PictureController

```

<?php

namespace App\Http\Controllers;

use Intervention\Image\ImageManagerStatic as Image;

use Illuminate\Support\Facades\File;

use Illuminate\Http\Request;

use App\Models\Picture;

use DateTime;

class PictureController extends Controller
{
    public function insertPicture(Request $request)
    {
        $rules =

```

```

[
    'pictureNameAdmin' => 'required|unique:picture,picture_name',
    'pictureFileAdmin' => 'required|mimes:jpg,jpeg',
    'pictureCategoryAdmin' => 'not_in:0', 'pictureUserAdmin' =>
    'not_in:0'
];

$messages =
[
    'required'           => 'Field :attribute is required!',
    'pictureNameAdmin.unique' => 'Picture name must be unique!',
    'mimes'              => 'Picture is not in valid format! Allowed formats are :values.',
    'pictureCategoryAdmin.not_in' => 'Category must be selected!',
    'pictureUserAdmin.not_in'    => 'User must be selected!'
];

$request->validate($rules, $messages);

$image = $request->file('pictureFileAdmin');
$extension = $image->getClientOriginalExtension();
$tmp_path = $image->getPathName();

$folder = 'images/fulls/';
$file_name = time()." ".$extension;
$new_path = public_path($folder).$file_name;

$thumb_image = Image::make($image->getRealPath());
$thumb_image->resize(450, 450);

try
{
    File::move($tmp_path, $new_path);
    $thumb_image->save(public_path('images/thumbs/'.$file_name));

    $picture = new Picture();

    $date = new DateTime();
    $date->format('d.m.Y');

    $picture->picture_name = $request->get('pictureNameAdmin');
    $picture->picture_show = 'images/thumbs/'.$file_name;
    $picture->picture_path = 'images/fulls/'.$file_name;
    $picture->category_id = $request->get('pictureCategoryAdmin');

```

```

    $picture->user_id = $request-
    >get('pictureUserAdmin'); $picture->shared_at = $date;
    $picture->updated_at = null;

    $picture->insert();

    return redirect('/admin-panel/pictures/')->with('success', 'Picture successfully added!');
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while adding picture!');
}
catch(\Symfony\Component\HttpFoundation\File\Exception\FileException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error with picture upload!');
}
}

public function updatePicture($id, Request $request)
{
    $rules =
    [
        'pictureNameAdmin'    => 'required',
        'pictureCategoryAdmin' => 'not_in:0',
        'pictureUserAdmin'    => 'not_in:0'
    ];

    $messages =
    [
        'required'            => 'Field :attribute is required!',
        'pictureCategoryAdmin.not_in' => 'Category must be selected!',

        'pictureUserAdmin.not_in'    => 'User must be selected!'
    ];

    $request->validate($rules, $messages);

    try
    {
        $picture = new Picture();

        $date = new DateTime();

```

```

$date->format('d.m.Y');

$picture->picture_id = $id;
$picture->picture_name = $request->get('pictureNameAdmin');
$picture->category_id = $request->get('pictureCategoryAdmin');
$picture->user_id = $request->get('pictureUserAdmin');
$picture->picture_show = $picture->get()->picture_show;
$picture->picture_path = $picture->get()->picture_path;
$picture->updated_at = $date;

$image = $request->file('pictureFileAdmin');

if(!empty($image))
{
    $picture_to_update = $picture->get();
    File::delete($picture_to_update->picture_path);
    File::delete($picture_to_update->picture_show);

    $extension = $image->getClientOriginalExtension();
    $tmp_path = $image->getPathName();

    $folder = 'images/fulls/';
    $file_name = time()." ".$extension;
    $new_path = public_path($folder).$file_name;

    $thumb_image = Image::make($image->getRealPath());
    $thumb_image->resize(450, 450);

    $picture->picture_show = 'images/thumbs/'.$file_name;
    $picture->picture_path = 'images/fulls/'.$file_name;

    File::move($tmp_path, $new_path);
    $thumb_image->save(public_path('images/thumbs/'.$file_name));
}

$picture->update();

return redirect('/admin-panel/pictures')->with('success', 'Picture successfully updated!');
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while updating picture!');
}

```

```

    }
    catch(\Symfony\Component\HttpFoundation\File\Exception\FileException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error with picture upload!');
    }
}

```

```

public function deletePicture($id)
{
    try
    {
        $picture = new Picture();
        $picture->picture_id = $id;

        $picture_to_delete = $picture->get();
        File::delete($picture_to_delete->picture_path);
        File::delete($picture_to_delete->picture_show);

        $picture->delete();

        return redirect()->back()->with('success', 'Picture successfully deleted!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while deleting picture!');
    }
}

```

```

public function sharePicture(Request $request)
{
    $rules =
    [
        'pictureName'    => 'required|unique:picture,picture_name',
        'pictureUpload'  => 'required|mimes:jpg,jpeg', => 'not_in:0'
        'pictureCategory'
    ];

    $messages =
    [
        'required'        => 'Field :attribute is required!',
        'pictureName.unique' => 'Picture name must be unique!',
    ]
}

```

```

        'mimes' => 'Picture is not in valid format! Allowed formats are :values.',
        'pictureCategory.not_in' => 'Category must be selected!'
    ];

    $request->validate($rules, $messages);

    $image = $request->file('pictureUpload');
    $extension = $image->getClientOriginalExtension();
    $tmp_path = $image->getPathName();

    $folder = 'images/fulls/';
    $file_name = time()." ".$extension;
    $new_path = public_path($folder).$file_name;

    $thumb_image = Image::make($image->getRealPath());
    $thumb_image->resize(450, 450);

    try
    {
        File::move($tmp_path, $new_path);
        $thumb_image->save(public_path('images/thumbs/'.$file_name));

        $picture = new Picture();

        $date = new DateTime();
        $date->format('d.m.Y');

        $picture->picture_name = $request->get('pictureName');
        $picture->picture_show = 'images/thumbs/'.$file_name;
        $picture->picture_path = 'images/fulls/'.$file_name;
        $picture->category_id = $request->get('pictureCategory');
        $picture->user_id = session()->get('user')[0]->user_id;
        $picture->shared_at = $date;
        $picture->updated_at = null;

        $picture->insert();

        return redirect('/share')->with('success', 'You have successfully shared picture!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while sharing picture!');
    }

```

```

    }
    catch(\Symfony\Component\HttpFoundation\File\Exception\FileException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error with picture upload!');
    }
}

public function deleteMyPicture($id, Request $request)
{
    if(session()->has('user'))
    {
        try
        {
            $picture = new Picture();

            $picture->user_id = $request->session()->get('user')[0]-
            >user_id; $picture->picture_id = $id;

            $picture->deleteMyPic();

            return redirect('/share')->with('success', 'You have successfully deleted your picture!');
        }
        catch(\Illuminate\Database\QueryException $ex)
        {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Error while deleting picture!');
        }
    }
    else
    {
        return redirect('401');
    }
}
}

```

## PollController

```
<?php
```

```
namespace App\Http\Controllers;
```



```

use Illuminate\Http\Request;

use App\Models\Poll;

use DateTime;

class PollController extends Controller
{
    public function showPoll()
    {
        try
        {
            $poll = new Poll();

            $myPoll = $poll->get();

            return response($myPoll, 200);
        }
        catch(\Exception $ex)
        {
            \Log::error($ex->getMessage());
            return response(null, 500);
        }
    }

    public function insertPoll(Request $request)
    {
        $request->validate([
            'pollQuestionAdmin' => 'required',
            'pollFirstAnswer'   => 'required',
            'pollSecondAnswer' => 'required'
        ],[
            'required' => 'Field :attribute is required!'
        ]);

        $question = $request->get('pollFirstAnswer');
        $answer = $request->get('pollFirstAnswer');
        $answer_alt = $request->get('pollSecondAnswer');

        try
        {
            $poll = new Poll();

```

```

$date = new DateTime();
$date->format('d.m.Y');

$poll->poll_question = $question;
$poll->answer = $answer;
$poll->answer_alt = $answer_alt;
$poll->created_at = $date;
$poll->updated_at = null;
$poll->poll_id = $poll->insertPoll();
$poll->insertFirstAnswer();
$poll->insertSecondAnswer();
return redirect('/admin-panel/polls')->with('success', 'Poll successfully added!');

}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while adding poll!');
}
}

public function updatePoll(Request $request, $id)
{
    $rules =
    [
        'pollQuestionAdmin' => 'required',
        'pollFirstAnswer' => 'required',
        'pollSecondAnswer' => 'required'
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!'
    ];

    $request->validate($rules, $messages);

    try
    {
        $poll = new Poll();

        $date = new DateTime();
        $date->format('d.m.Y');
    }
}

```

```

    $poll->poll_id = $id;
    $poll->poll_question = $request->get('pollQuestionAdmin');
    $poll->answer = $request->get('pollFirstAnswer');
    $poll->answer_alt = $request->get('pollSecondAnswer');
    $poll->updated_at = $date;
    $poll->answer_id = $request->get('answer');
    $poll->answer_id_alt = $request->get('answer_alt');

    $poll->updateFirstAnswer();
    $poll->updateSecondAnswer();

    $poll->updatePoll();

    return redirect('/admin-panel/polls')->with('success', 'Poll updated successfully!');

}
catch(\Exception $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while updating poll!');
}
}

public function deletePoll($id)
{
    try
    {
        $poll = new Poll();

        $poll->poll_id = $id;

        $poll->deleteVote();

        $poll->deleteAnswer();

        $poll->deletePoll();

        return redirect('/admin-panel/polls')->with('success', 'Poll successfully deleted!');
    }
    catch(\Exception $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

        return redirect()->back()->with('error', 'Error while deleting poll!');
    }
}

public function insertVote(Request $request)
{
    $poll = new Poll();

    $poll->user_id = $request->get('user_id');
    $poll->poll_id = $request->get('poll_id');
    $poll->answer_id = $request->get('answer_id');

    try
    {
        $result = $poll->insertVote();

        $poll->incAnswer();

        return response($result, 200);
    }
    catch (\Exception $ex)
    {
        \Log::error($ex->getMessage());
        return response(null, 500);
    }
}

public function activatePoll($id)
{
    try
    {
        $poll = new Poll();

        $poll->poll_id = $id;

        $poll->deactivate();
        $poll->activate();

        return redirect('/admin-panel/polls')->with('success', 'Poll activated successfully!');
    }
    catch (\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
    }
}

```

```

        return redirect()->back()->with('error', 'Error while activating poll!');
    }
}
}

```

## RoleController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Role;

use DateTime;

class RoleController extends Controller
{
    public function insertRole(Request $request)
    {
        $rules =
        [
            'roleNameAdmin' => 'required|unique:role,role_name'
        ];

        $messages =
        [
            'required' => 'Field :attribute is required!',
            'roleNameAdmin.unique' => 'Role name must be unique!',
        ];

        $request->validate($rules, $messages);

        try
        {
            $role = new Role();

            $date = new DateTime();
            $date->format('d.m.Y');

            $role->role_name = $request->get('roleNameAdmin');

```

```

    $role->created_at = $date;
    $role->updated_at = null;

    $role->insert();

    return redirect('/admin-panel/roles')->with('success', 'Role successfully added!');
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while adding role!');
}
}

public function updateRole($id, Request $request)
{
    $rules =
    [
        'roleNameAdmin' => 'required'
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!',
    ];

    $request->validate($rules, $messages);

    try
    {
        $role = new Role();

        $date = new DateTime();
        $date->format('d.m.Y');

        $role->role_id = $id;
        $role->role_name = $request-
        >get('roleNameAdmin'); $role->updated_at = $date;

        $role->update();

        return redirect('/admin-panel/roles')->with('success', 'Role successfully updated!');
    }
}

```

```

        catch(Illuminate\Database\QueryException $ex)
        {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Error while updating role!');
        }
    }

    public function deleteRole($id)
    {
        try
        {
            $role = new Role();

            $role->role_id = $id;

            $role->delete();

            return redirect('/admin-panel/roles')->with('success', 'Role successfully deleted!');
        }
        catch(Illuminate\Database\QueryException $ex)
        {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Error while deleting role!');
        }
    }
}

```

## RoleController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\User;

use DateTime;

class UserController extends Controller
{
    public function login(Request $request)
    {

```

```

$rules =
[
    'usernameLogin' => 'required',
    'passwordLogin' => 'required'
];

$messages =
[
    'required' => 'Field :attribute is required!'
];

$request->validate($rules, $messages);

$user = new User();

$user->user_name = $request->get('usernameLogin');
$user->user_pass = $request->get('passwordLogin');

$userLogin = $user->getUser();

if(!empty($userLogin))
{
    $request->session()->push('user', $userLogin);

    if(session()->get('user')[0]->role_name == 'administrator')
    {
        return redirect('/admin-panel/users');
    }
    elseif(session()->get('user')[0]->role_name == 'user')
    {
        return redirect('/share');
    }
}
else
{
    return redirect('/home')->with('error', 'There is no user with that username registered!');
}
}

public function logout(Request $request)
{
    $request->session()->forget('user');
    $request->session()->flush();
}

```



```

        return redirect('/home');
    }

    public function register(Request $request)
    {
        $rules = [
            'emailRegister' => ['regex:/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/','required','unique:user,user_mail'],
            'usernameRegister' => ['regex:/^[a-zA-Z0-9]{3,15}$/','required','unique:user,user_name','min:3','max:15'],
            'passwordRegister' => ['regex:/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{6,}$/','required','confirmed','min:6']
        ];

        $messages = [
            'required'          => 'Field :attribute is required!',
            'confirmed'         => 'Passwords do not match!',
            'emailRegister.unique' => 'E-mail is already registered!',
            'usernameRegister.unique' => 'Username is already taken!',
            'emailRegister.regex'   => 'E-mail is not in valid format!',
            'usernameRegister.regex' => 'Username is not in valid format!',
            'passwordRegister.regex' => 'Password is not in valid format!'
        ];

        $request->validate($rules, $messages);

        try
        {
            $user = new User();

            $date = new DateTime();
            $date->format('d.m.Y');

            $user->user_mail = $request->get('emailRegister');
            $user->user_name = $request->get('usernameRegister');
            $user->user_pass = md5($request->get('passwordRegister')); $user->role_id = 2;
            $user->registered_at = $date;
            $user->changed_at = null;

            $user->insert();

            return redirect('/home')->with('success', 'You have successfully registered!');
        }
    }
}

```

```

    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error with registering!');
    }
}

public function insertUser(Request $request)
{
    $rules = [
        'userEmailAdmin'    => 'required|unique:user,user_mail',
        'userUsernameAdmin' => 'required|unique:user,user_name',
        'userPasswordAdmin' => 'required',
        'userRoleAdmin'     => 'not-in:0'
    ];

    $messages = [
        'required'           => 'Field :attribute is required!',
        'userEmailAdmin.unique' => 'E-mail must be unique!',
        'userUsernameAdmin.unique' => 'Username must be unique!',
        'emailRegister.regex'    => 'E-mail is not in valid format!',
        'usernameRegister.regex' => 'Username is not in valid format!',
        'passwordRegister.regex' => 'Password is not in valid format!',
        'userRoleAdmin.not_in'   => 'Role must be selected!'
    ];

    $request->validate($rules, $messages);

    try
    {
        $user = new User();

        $date = new DateTime();
        $date->format('d.m.Y');

        $user->user_mail = $request->get('userEmailAdmin');
        $user->user_name = $request->get('userUsernameAdmin');
        $user->user_pass = md5($request->get('userPasswordAdmin'));
        $user->role_id = $request->get('userRoleAdmin');
        $user->registered_at = $date;
        $user->changed_at = null;
    }
}

```

```

        $user->insert();

        return redirect('/admin-panel/users')->with('success', 'User successfully added!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while adding user!');
    }
}

public function updateUser($id, Request $request)
{
    $rules =
    [
        'userEmailAdmin'      => 'required',
        'userUsernameAdmin'   => 'required',
        'userPasswordAdmin'   => 'required',
        'userRoleAdmin'       => 'not-in:0'
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!',
        'userEmailAdmin.unique' => 'E-mail must be unique!',
        'userUsernameAdmin.unique' => 'Username must be unique!',
        'emailRegister.regex' => 'E-mail is not in valid format!',
        'usernameRegister.regex' => 'Username is not in valid format!',
        'passwordRegister.regex' => 'Password is not in valid format!',
        'userRoleAdmin.not_in' => 'Role must be selected!'
    ];

    $request->validate($rules, $messages);

    try
    {
        $user = new User();

        $date = new DateTime();
        $date->format('d.m.Y');

        $user->user_id = $id;
        $user->user_mail = $request->get('userEmailAdmin');
    }
}

```

```

$user->user_name = $request-
>get('userUsernameAdmin'); $user->role_id = $request-
>get('userRoleAdmin'); $user->changed_at = $date;

$pass = $request->get('userPasswordAdmin');

if(strlen($pass) == 32)
{
    $user->user_pass = $request->get('userPasswordAdmin');
}
else
{
    $user->user_pass = md5($request->get('userPasswordAdmin'));
}

$user->update();

return redirect('/admin-panel/users')->with('success', 'User successfully updated!');
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while updating user!');
}
}

public function deleteUser($id)
{
    try
    {
        $user = new User();

        $user->user_id = $id;

        $user->delete();

        return redirect('/admin-panel/users')->with('success', 'User successfully deleted!');
    }
    catch(\Illuminate\Database\QueryException $ex)
    {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Error while deleting user!');
    }
}

```

```

}

public function changeUser($id, Request $request)
{
    $rules =
    [
        'userEmailChange'    => ['regex:/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/','required'],
        'userUsernameChange' => ['regex:/^[a-zA-Z0-9]{3,15}$/','required','min:3','max:15'],
        'userPasswordChange' => ['regex:/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{6,}$/','required','min:6']
    ];

    $messages =
    [
        'required' => 'Field :attribute is required!', 'userEmailChange.regex'
        => 'E-mail is not in valid format!', 'userUsernameChange.regex' =>
        'Username is not in valid format!', 'userPasswordChange.regex' =>
        'Password is not in valid format!',
    ];

    $request->validate($rules, $messages);

    try
    {
        $user = new User();

        $date = new DateTime();
        $date->format('d.m.Y');

        $user->user_id = $id;
        $user->user_mail = $request->get('userEmailChange');
        $user->user_name = $request->get('userUsernameChange'); $user->changed_at = $date;

        $pass = $request->get('userPasswordChange');

        if(strlen($pass) == 32)
        {
            $user->user_pass = $request->get('userPasswordChange');
        }
        else
        {

```

```

        $user->user_pass = md5($request->get('userPasswordChange'));
    }

    $result = $user->change();

    if($result == 1)
    {

        return redirect('/share')->with('success', 'You have successfully updated your profile!');
    }
    else
    {
        abort(404);
    }
}
catch(\Illuminate\Database\QueryException $ex)
{
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Error while updating profile!');
}
}
}

```

## Middleware/AdminMiddleware

```
<?php

namespace App\Http\Middleware;

use Closure;

class AdminMiddleware
{
    public function handle($request, Closure $next)
    {
        if(session()->has('user'))
        {
            $user = session()->get('user')[0];

            if($user->role_name == 'administrator')
            {
                return $next($request);
            }
            else
            {
                return redirect('/home');
            }
        }
        else
        {
            return redirect('/home');
        }
    }
}
```

## Middleware/UserMiddleware

```
<?php

namespace App\Http\Middleware;

use Closure;

class UserMiddleware
{
    public function handle($request, Closure $next)
```

```
{
  if(session()->has('user'))
  {
    return $next($request);
  }
  else
  {
    return redirect('/home');
  }
}
```



## Models/Category.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Category {

    public $category_id;
    public $category_name;
    public $created_at;
    public $updated_at;

    public function getAll()
    {
        $result = DB::table('category')
            ->select('*')
            ->get();

        return $result;
    }

    public function get()
    {
        $result = DB::table('category')
            ->select('*')
            ->where('category_id',$this->category_id)
            ->first();

        return $result;
    }

    public function insert()
    {
        $result = DB::table('category')
            ->insert([
                'category_name' => $this->category_name,
                'created_at'    => $this->created_at,
                'updated_at'    => $this->updated_at
            ]);
    }
}
```

```

        return $result;
    }

    public function update()
    {
        $result = DB::table('category')
            ->where('category_id',$this->category_id)
            ->update([
                'category_name' => $this->category_name,
                'updated_at'    => $this->updated_at
            ]);

        return $result;
    }

    public function delete()
    {
        $result = DB::table('category')
            ->where('category_id',$this->category_id)
            ->delete();

        return $result;
    }
}

```

## Models/Comment.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Comment {

    public $comment_id;
    public $comment_text;
    public $picture_id;
    public $user_id;
    public $created_at;
    public $updated_at;

    public $uid;

    public function getByPost($id)
    {
        $result = DB::table('comment')
            ->select('*')
            ->join('user','user.user_id','=','comment.user_id')
            ->where('comment.picture_id','=',$id)
            ->orderBy('created_at','asc')
            ->get();

        return $result;
    }

    public function get()
    {
        $result = DB::table('comment')
            ->select('*')
            ->where('comment_id',$this->comment_id)
            ->first();

        return $result;
    }

    public function insert()
    {
```

```

$result = DB::table('comment')
->insert([
    'comment_text' => $this->comment_text,
    'picture_id' => $this->picture_id,
    'user_id' => $this->user_id,
    'created_at' => $this->created_at,
    'updated_at' => $this->updated_at
]);

return $result;
}

public function update()
{
    $result = DB::table('comment')
        ->where('comment_id',$this->comment_id)
        ->update([
            'comment_text' => $this->comment_text,
            'updated_at' => $this->updated_at
        ]);

    return $result;
}

public function deleteAdminComment()
{
    $result = DB::table('comment')
        ->where('comment_id',$this->comment_id)
        ->delete();

    return $result;
}

public function deleteComment()
{
    return DB::table('comment')
        ->where([
            ['comment_id',$this->comment_id],
            ['user_id',$this->user_id]
        ])
        ->delete();
}
}

```

## Models/Navigation.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Navigation {

    public $navigation_id;
    public $navigation_name;
    public $navigation_path;
    public $navigation_icon;
    public $created_at;
    public $updated_at;

    public function getAll()
    {
        $result = DB::table('navigation')
            ->select('*')
            ->get();

        return $result;
    }

    public function get()
    {
        $result = DB::table('navigation')
            ->select('*')
            ->where('navigation_id',$this->navigation_id)
            ->first();

        return $result;
    }

    public function getAuth()
    {
        $result = DB::table('navigation')
            ->select('*')
            ->whereNotIn('navigation_id', [4])
            ->get();
    }
}
```

```

        return $result;
    }

    public function getNoAuth()
    {
        $result = DB::table('navigation')
            ->select('*')
            ->whereNotIn('navigation_id', [3,4,6])
            ->get();

        return $result;
    }

    public function insert()
    {
        $result = DB::table('navigation')
            ->insert([
                'navigation_name' => $this->navigation_name,
                'navigation_path' => $this->navigation_path,
                'navigation_icon' => $this->navigation_icon,
                'created_at'      => $this->created_at,
                'updated_at'      => $this->updated_at
            ]);

        return $result;
    }

    public function update()
    {
        $result = DB::table('navigation')
            ->where('navigation_id',$this->navigation_id)
            ->update([
                'navigation_name' => $this->navigation_name,
                'navigation_path' => $this->navigation_path,
                'navigation_icon' => $this->navigation_icon,
                'updated_at'      => $this->updated_at
            ]);

        return $result;
    }

    public function delete()
    {

```

```

        $result = DB::table('navigation')
            ->where('navigation_id',$this->navigation_id)
            ->delete();

        return $result;
    }
}

```

## Models/Picture.php

```

<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Picture {

    public $picture_id;
    public $picture_name;
    public $picture_show;
    public $picture_path;
    public $category_id;
    public $user_id;
    public $shared_at;
    public $updated_at;

    public function getAll()
    {
        $result = DB::table('picture')
            ->select('*', 'picture.updated_at as pictureUpdated')
            ->join('category', 'category.category_id', '=', 'picture.category_id')
            ->join('user', 'user.user_id', '=', 'picture.user_id')
            ->orderBy('picture_id')
            ->get();

        return $result;
    }

    public function getLimited()
    {
        $result = DB::table('picture')

```

```

        ->select('*')
        ->orderBy('picture_id','desc')
        ->limit(8)
        ->get();

    return $result;
}

public function getCategorized($cat)
{
    $result = DB::table('picture')
        ->select('*')
        ->join('category','category.category_id','=','picture.category_id')
        ->join('user','user.user_id','=','picture.user_id')
        ->where('picture.category_id','=',$cat)
        ->orderBy('picture_id')
        ->get();

    return $result;
}

public function getByUser($id)
{
    $result = DB::table('picture')
        ->select('*')
        ->join('category','category.category_id','=','picture.category_id')
        ->join('user','user.user_id','=','picture.user_id')
        ->where('picture.user_id','=',$id)
        ->orderBy('shared_at')
        ->get();

    return $result;
}

public function getSingle($id)
{
    $result = DB::table('picture')
        ->select('*','\DB::raw("(SELECT count(views_id) FROM views WHERE picture_id
= $id) as views)")
        ->join('category','category.category_id','=','picture.category_id')
        ->join('user','user.user_id','=','picture.user_id')
        ->where('picture.picture_id','=',$id)
        ->first();

```



```

        return $result;
    }

    public function get()
    {
        $result = DB::table('picture')
            ->select('*')
            ->where('picture_id',$this->picture_id)
            ->first();

        return $result;
    }

    public function insert()
    {
        $result = DB::table('picture')
            ->insert([
                'picture_name' => $this->picture_name,
                'picture_show' => $this->picture_show,
                'picture_path' => $this->picture_path,
                'category_id' => $this->category_id,
                'user_id'      => $this->user_id,
                'shared_at'    => $this->shared_at,
                'updated_at'   => $this->updated_at
            ]);

        return $result;
    }

    public function update()
    {
        $result = DB::table('picture')
            ->where('picture_id',$this->picture_id)
            ->update([
                'picture_name' => $this->picture_name,
                'picture_show' => $this->picture_show,
                'picture_path' => $this->picture_path,
                'category_id' => $this->category_id,
                'user_id'      => $this->user_id,
                'updated_at'   => $this->updated_at
            ]);
    }

```

```

        return $result;
    }

    public function delete()
    {
        $result = DB::table('picture')
            ->where('picture_id',$this->picture_id)
            ->delete();

        return $result;
    }

    public function deleteMyPic()
    {
        $result = DB::table('picture')
            ->where([
                ['picture.user_id',$this->user_id],
                ['picture.picture_id',$this->picture_id]
            ])
            ->delete();

        return $result;
    }

    public function getViews($id, $ip)
    {
        $result = DB::table('views')
            ->insert([
                'ip' => $ip,
                'picture_id' => $id
            ]);

        return $result;
    }
}

```

## Models/Poll.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Poll {

    public $poll_id;
    public $user_id;
    public $answer_id;
    public $answer_id_alt;
    public $poll_question;
    public $poll_active;
    public $created_at;
    public $updated_at;
    public $answer;
    public $answer_alt;

    public function getAll()
    {
        $result = DB::table('poll')
            ->select('*')
            ->get();

        return $result;
    }

    public function getOne()
    {
        $result = DB::table('poll')
            ->select('*')
            ->join('answer','answer.poll_id','=','poll.poll_id')
            ->where('poll.poll_id','=',$this->poll_id)
            ->get();

        return $result;
    }

    public function get()
    {

```

```

$result = DB::table('poll')
    ->select('*')
    ->join('answer','answer.poll_id','=', 'poll.poll_id')
    ->where('poll_active','=',1)
    ->get();

return $result;
}

public function findUser()
{
    $result = DB::table('vote')
        ->join('poll','poll.poll_id','=', 'vote.poll_id')
        ->where([
            ['vote.user_id','=', $this->user_id],
            ['poll.poll_active','=',1]
        ])
        ->first();

    return $result;
}

public function insertPoll()
{
    $result = DB::table('poll')
        ->insertGetId([
            'poll_question' => $this->poll_question,
            'poll_active' => 0,
            'created_at' => $this->created_at
        ]);

    return $result;
}

public function updatePoll()
{
    $result = DB::table('poll')
        ->where('poll_id','=', $this->poll_id)
        ->update([
            'poll_question' => $this->poll_question,
            'updated_at' => $this->updated_at
        ]);
}

```

```

        return $result;
    }

    public function deletePoll()
    {
        return DB::table('poll')
            ->where('poll_id','=', $this->poll_id)
            ->delete();
    }

    public function insertFirstAnswer()
    {
        $result = DB::table('answer')
            ->insert([
                'poll_id'    => $this->poll_id,
                'answer'      => $this->answer,
                'answer_votes' => 0
            ]);

        return $result;
    }

    public function insertSecondAnswer()
    {
        $result = DB::table('answer')
            ->insert([
                'poll_id'    => $this->poll_id,
                'answer'      => $this->answer_alt,
                'answer_votes' => 0
            ]);

        return $result;
    }

    public function updateFirstAnswer()
    {
        $result = DB::table('answer')
            ->where([
                'poll_id' => $this->poll_id,
                'answer_id' => $this->answer_id
            ])
            ->update([

```

```

        'answer' => $this->answer
    ]);

    return $result;
}

public function updateSecondAnswer()
{
    $result = DB::table('answer')
        ->where([
            'poll_id' => $this->poll_id,
            'answer_id' => $this->answer_id_alt
        ])
        ->update([
            'answer' => $this->answer_alt
        ]);

    return $result;
}

public function deleteAnswer()
{
    $result = DB::table('answer')
        ->where('poll_id','=', $this->poll_id)
        ->delete();

    return $result;
}

public function incAnswer()
{
    $result = DB::table('answer')
        ->where('answer_id','=', $this->answer_id)
        ->increment('answer_votes',1);

    return $result;
}

public function insertVote()
{
    $result = DB::table('vote')
        ->insertGetId([
            'poll_id' => $this->poll_id,

```

```

        'user_id' => $this->user_id
    );

    return $result;
}

public function deleteVote()
{
    $result = DB::table('vote')
        ->where('poll_id','=', $this->poll_id)
        ->delete();

    return $result;
}

public function activate()
{
    $result = DB::table('poll')
        ->where('poll_id','=', $this->poll_id)
        ->update([
            'poll_active' => 1
        ]);

    return $result;
}

public function deactivate()
{
    $result = DB::table('poll')
        ->update([
            'poll_active' => 0
        ]);

    return $result;
}
}

```

## Models/Role.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Role {

    public $role_id;
    public $role_name;
    public $created_at;
    public $updated_at;

    public function getAll()
    {
        $result = DB::table('role')
            ->select('*')
            ->get();

        return $result;
    }

    public function get()
    {
        $result = DB::table('role')
            ->select('*')
            ->where('role_id',$this->role_id)
            ->first();

        return $result;
    }

    public function insert()
    {
        $result = DB::table('role')
            ->insert([
                'role_name' => $this->role_name,
                'created_at' => $this->created_at,
                'updated_at' => $this->updated_at
            ]);
    }
}
```



```

        return $result;
    }

    public function update()
    {
        $result = DB::table('role')
            ->where('role_id',$this->role_id)
            ->update([
                'role_name' => $this->role_name,
                'updated_at' => $this->updated_at
            ]);

        return $result;
    }

    public function delete()
    {
        $result = DB::table('role')
            ->where('role_id',$this->role_id)
            ->delete();

        return $result;
    }
}

```

## Models/User.php

```
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class User {

    public $user_id;
    public $user_mail;
    public $user_name;
    public $user_pass;
    public $role_id;
    public $registered_at;
    public $changed_at;

    public function getAll()
    {
        $result = DB::table('user')
            ->select('*')
            ->join('role','role.role_id','=','user.role_id')
            ->get();

        return $result;
    }

    public function getUser()
    {
        $result = DB::table('user')
            ->select('user.*','role.role_name')
            ->join('role','role.role_id','=','user.role_id')
            ->where([
                'user_name' => $this->user_name,
                'user_pass' => md5($this->user_pass)
            ])
            ->first();

        return $result;
    }

    public function getProfile($id)
```

```

{
    $result = DB::table('user')
        ->select('*')
        ->where('user_id','=', $id)
        ->first();

    return $result;
}

public function get()
{
    $result = DB::table('user')
        ->select('*')
        ->where('user_id', $this->user_id)
        ->first();

    return $result;
}

public function insert()
{
    $result = DB::table('user')
        ->insert([
            'user_mail' => $this->user_mail,
            'user_name' => $this->user_name,
            'user_pass' => $this->user_pass,
            'role_id' => $this->role_id,
            'registered_at' => $this->registered_at,
            'changed_at' => $this->changed_at
        ]);

    return $result;
}

public function update()
{
    $result = DB::table('user')
        ->where('user_id', $this->user_id)
        ->update([
            'user_mail' => $this->user_mail,
            'user_name' => $this->user_name,
            'user_pass' => $this->user_pass,
            'role_id' => $this->role_id,

```

```

        'changed_at' => $this->changed_at
    );

    return $result;
}

public function delete()
{
    $result = DB::table('user')
        ->where('user_id',$this->user_id)
        ->delete();

    return $result;
}

public function change()
{
    $result = DB::table('user')
        ->where('user_id',$this->user_id)
        ->update([
            'user_mail' => $this->user_mail,
            'user_name' => $this->user_name,
            'user_pass' => $this->user_pass,
            'changed_at' => $this->changed_at
        ]);

    return $result;
}
}

```

## layout/template.blade.php

```
<title>@yield('title')</title>

@include('components.navigation')

@if(session()->has('user'))
    <div class="login-message">
        <h3>Logged in as: {{ session()->get('user')[0]->user_name }}</h3>
        <div class="arrow-down"></div>
    </div>
@endif

@yield('center')

        @isset($errors)
            @if($errors->any())
                @foreach($errors->all() as $error)
                    <div class="error"><i class="fa fa-exclamation-triangle"></i><p>{{
$error }}</p><i class="fa fa-times-circle close"></i></div>
                @endforeach
            @endif
        @endisset

        @empty(!session('error'))
            <div class="error"><i class="fa fa-exclamation-triangle"></i><p>{{
session('error') }}</p><i class="fa fa-times-circle close"></i></div>
        @endempty

        @empty(!session('success'))
            <div class="success"><i class="fa fa-thumbs-up"></i><p>{{ session('success')
}}</p><i class="fa fa-times-circle close"></i></div>
        @endempty

@include('components.footer')

@section('scripts')

@show
```

## components/navigation.blade.php

```
@if(session()->has('user'))
    @if(session()->get('user')[0]->role_name == 'administrator')
        @isset($navigations)
            @foreach($navigations as $navigation)
                <li><a href="{{ asset($navigation->navigation_path) }}" class="{{
Request::is($navigation->navigation_path) ? 'active' : null }}"><span class="icon
{{ ($navigation->navigation_icon) }}"></span></a></li>
            @endforeach
        @endisset
    @elseif(session()->get('user')[0]->role_name == 'user')
        @isset($authNavigations)
            @foreach($authNavigations as $authNavigation)
                <li><a href="{{ asset($authNavigation->navigation_path) }}" class="{{
Request::is($authNavigation->navigation_path) ? 'active' : null }}"><span class="icon {{
($authNavigation->navigation_icon) }}"></span></a></li>
            @endforeach
        @endisset
    @endif
@else
    @isset($noAuthNavigations)
        @foreach($noAuthNavigations as $noAuthNavigation)
            <li><a href="{{ asset($noAuthNavigation->navigation_path) }}" class="{{
Request::is($noAuthNavigation->navigation_path) ? 'active' : null }}"><span class="icon
{{ ($noAuthNavigation->navigation_icon) }}"></span></a></li>
        @endforeach
    @endisset
@endif
```

## components/footer.blade.php

```
@isset($navigations)
    @foreach($navigations as $navigation)
        <li><a href="{{ asset($navigation->navigation_path) }}" class="{{ ($loop->first)?
'active' : '' }}">{{ ($navigation->navigation_name) }}</a></li>
    @endforeach
@endisset
```

## pages/admin.blade.php

```
@extends('layout.template')

@section('title')
    Snapshot | Admin Panel
@endsection

@section('center')

    <li><a href="{{ asset('/admin-panel/users') }}" class="button {{ Request::is('admin-
panel/users') ? 'active' : null }}" id="buttonUsers">USERS</a></li>
    <li><a href="{{ asset('/admin-panel/roles') }}" class="button {{ Request::is('admin-
panel/roles') ? 'active' : null }}" id="buttonRoles">ROLES</a></li>
    <li><a href="{{ asset('/admin-panel/pictures') }}" class="button {{ Request::is('admin-
panel/pictures') ? 'active' : null }}" id="buttonPictures">PICTURES</a></li>
    <li><a href="{{ asset('/admin-panel/categories') }}" class="button {{ Request::is('admin-
panel/categories') ? 'active' : null }}" id="buttonCategories">CATEGORIES</a></li>
    <li><a href="{{ asset('/admin-panel/navigation') }}" class="button
{{ Request::is('admin-panel/navigation') ? 'active' : null }}"
id="buttonNavigation">NAVIGATION</a></li>
    <li><a href="{{ asset('/admin-panel/polls') }}" class="button {{ Request::is('admin-
panel/polls') ? 'active' : null }}" id="buttonPolls">POLL</a></li>

@yield('admin')

@endsection

@section('scripts')

@parent

@endsection
```

## pages/gallery.blade.php

```
@extends('layout.template')

@section('title')
    Snapshot - Gallery
@endsection

@section('center')

    @isset($categories)
        @foreach($categories as $category)
            <li><a href="{{ asset('/gallery/'.$category->category_id) }}" class="button"
id="{{ { 'cat'.$category->category_id } }}">{{ $category->category_name }}</a></li>
        @endforeach
    @endisset

    @isset($pictures)
        @foreach($pictures as $picture)
            <div class="media">
                <a href="{{ asset('/pictures/'.$picture->picture_id) }}">picture_name) }}" /></a>
            </div>
        @endforeach
    @endisset

@endsection

@section('scripts')

    @parent

@endsection
```

## pages/home.blade.php

```
@extends('layout.template')

@section('title')
    Snapshot - Home
@endsection

@section('center')
```



```

        @isset($pictures)
            @foreach($pictures as $picture)
                <div class="media">
                    <a href="{{ asset('/pictures/'.$picture->picture_id ) }}">picture_name) }}" /></a>
                </div>
            @endforeach
        @endisset

<a href="{{ asset('/gallery') }}" class="button big">Full Gallery</a>

```

```

@if(session()->has('user'))
    <input type="hidden" value="{{ session()->get('user')[0]->user_id }}"
id="idUser"></input>
    @isset($result)
        <input type="hidden" value="1" id="exist"/>
    @else
        <input type="hidden" value="0" id="exist"/>
    @endisset
@else

    <form action="{{ route('login') }}" method="post">
        {{ csrf_field() }}

        <form action="{{ route('register') }}" method="post"
id="formRegister"> {{ csrf_field() }}

```

```
@endif
```

```
@endsection
```

```
@section('scripts')
```

```
    @parent
```

```
@endsection
```

## pages/information.blade.php

```
@extends('layout.template')
```

```
@section('title')
```

```
    Snapshot - Information
```

```

@endsection

@section('center')

    <form action="{{ route('mail') }}" method="post">
        {{ csrf_field()

}} @endsection

```

## pages/picture.blade.php

```

@extends('layout.template')

@section('title')
    Snapshot - Picture
@endsection

@section('center')

    @isset($singlePicture)
        <div class="picture_wrapper">
            picture_name) }}" />
            <h1>{{ ($singlePicture->picture_name)
}}</h1> <div class="picture-info">
                <div class="picture-left">
                    <h3>Shared: {{ date('d-M-Y', strtotime($singlePicture->created_at)) }}</h3>
                    <h3>By: {{ ($singlePicture->user_name) }}</h3>
                    <h3>Category: {{ ($singlePicture->category_name) }}</h3>
                    <h3>Views: {{ ($singlePicture->views) }}</h3>
                    <a href="{{ asset('/gallery') }}" class="button cancel">BACK</a>
                </div>
            @isset($comments)
                @if(count($comments) > 0 )
                    <div class="comments">
                        <h2>Comments</h2>
                        <div class="comment">
                            @foreach($comments as $comment)
                                <h4>{{ $comment->comment_text }}</h4>
                                <h5>{{ $comment->user_name }} said this - {{ date("d.m.Y
H:i", strtotime($comment->created_at)) }}
                                    @if(session()->has('user'))
                                        @if(session()->get('user')[0]->user_id == $comment->user_id)

```

```

        <a href="{{ asset('/pictures/'.$singlePicture-
>picture_id.'/'.$comment->comment_id) }}" class="inline"><i class="fa fa-edit inline"></i></a>
        <a href="{{ asset('/pictures/'.$singlePicture-
>picture_id.'/delete/'.$comment->comment_id) }}" class="inline"><i class="fa fa-
trash"></i></a>

        @elseif(session()->get('user')[0]->role_id == 1)
        <a href="{{ asset('/pictures/'.$singlePicture-
>picture_id.'/'.$comment->comment_id) }}" class="inline"><i class="fa fa-edit"></i></a>
        <a href="{{ asset('/pictures/'.$singlePicture-
>picture_id.'/delete/'.$comment->comment_id) }}" class="inline"><i class="fa fa-
trash"></i></a>

        @endif
    @endif
</h5>
    @endforeach
</div>
    @if(session()->has('user'))
        <form action="{{ (isset($selectedComment)) ?
asset('/pictures/'.$singlePicture->picture_id.'/update/'.$selectedComment->comment_id) :
asset('/pictures/'.$singlePicture->picture_id.'/comment') }}" method="post">
            {{ csrf_field() }}
            <textarea name="comment" placeholder="Write your comment">{{
(isset($selectedComment)) ? $selectedComment->comment_text : old('comment') }}</textarea>
            <div class="form-group">
                <input type="submit" name="btn-comment" value="{{
(isset($selectedComment)) ? 'UPDATE COMMENT' : 'COMMENT' }}" class="btn btn-
default"/>

                @isset($selectedComment)
                    <a href="{{ asset('pictures/'.$singlePicture->picture_id)
}}" class="button cancel">CANCEL</a>
                @endisset
            </div>
        </form>
    @endif
</div>
    @else
        <div class="comments">
            <h2>Comments</h2>
            <div class="comment">
                <div class="no-comments">
                    <h1>There are no comments on this picture. Be the first one to
comment!</h1>
                </div>
            </div>
        </div>
    @endif
</div>

```

```

        </div>
        @if(session()->has('user'))
            <form action="{{ (isset($selectedComment)) ?
asset('/pictures/'.$singlePicture->picture_id.'/update/'.$selectedComment->comment_id) :
asset('/pictures/'.$singlePicture->picture_id.'/comment') }}" method="post">
                {{ csrf_field() }}

                <textarea name="comment" placeholder="Write your comment"
value="{{ (isset($selectedComment)) ? $selectedComment->comment_text : old('comment')
}}"></textarea>

                <div class="form-group">
                    <input type="submit" name="btn-comment" value="{{
(isset($selectedComment)) ? 'UPDATE COMMENT' : 'COMMENT' }}" class="btn btn-
default"/>

                    @isset($selectedComment)
                        <a href="{{ asset('pictures/'.$singlePicture->picture_id)
}}" class="button cancel">CANCEL</a>
                    @endisset
                </div>
            </form>
        @endif
    </div>
    @endif
    @endisset
</div>
</div>
    @endisset
</div>
    @endisset
</div>
    @endsection

```

## pages/share.blade.php

```
@extends('layout.template')

@section('title')
    Snapshot - Share
@endsection

@section('center')

    <form action="{{ asset('/share/share') }}" method="post" enctype="multipart/form-data">
        {{ csrf_field() }}

        @isset($categories)
            @foreach($categories as $category)
                <option value="{{ $category->category_id }}">{{ $category->category_name
            }}</option>
            @endforeach
        @endisset

        @isset($pictures)
            @if($pictures->isEmpty())
                <h4>YOU DON'T HAVE ANY PICTURES. TRY TO SHARE ONE!</h4>
            @else
                <div class="userPicturesWrapper">
                    @foreach($pictures as $picture)
                        <div class="overlay">
                            picture_name) }}" />
                            <a href="{{ asset('/share/delete/'.$picture->picture_id) }}" class="icon fa-trash
middle"><span class="label">Delete</span></a>
                        </div>
                    @endforeach
                </div>
            @endif
        @endisset

        @isset($loggedUsers)
            <form action="{{ asset('/share/change/'.$loggedUsers->user_id) }}"
                method="post"> {{ csrf_field() }}

            <input type="email" name="userEmailChange" id="userEmailAdmin" value="{{ {
$loggedUsers->user_mail } }}" />
```

```
<input type="text" name="userUsernameChange" id="userUsernameAdmin" value="{{  
$loggedUsers->user_name }}" />
```

```
<input type="password" name="userPasswordChange" id="userPasswordAdmin"  
value="{{ $loggedUsers->user_pass }}" />
```

```
@endisset
```

```
@endsection
```

## admin/adminCategory.blade.php

```
@extends('pages.admin')

@section('admin')

    @isset($categories)
        @if(count($categories) > 5)
            <div class="wrapper-paging">
                <ul>
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li> <li><a
                    class="paging-this"><strong>0</strong> of <span>x</span></a></li> <li><a
                    class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
                </ul>
            </div>
        @endif
    @endisset

    @isset($categories)
        @foreach($categories as $category)
            <tr>
                <td>{{ $category->category_id }}</td>
                <td>{{ $category->category_name }}</td>
                <td>{{ date('d.m.Y', strtotime($category->created_at)) }}</td>
                <td>{{ (isset($category->updated_at)) ? date('d.m.Y',
                strtotime($category->updated_at)) : 'NEVER' }}</td>
                <td><a href="{{ { asset('/admin-panel/categories'.'/'. $category->category_id)
                }}"><i class="fa fa-edit"></i></a></td>
                <td><a href="{{ { asset('/admin-panel/categories/delete/'.' $category-
                >category_id) }}"><i class="fa fa-trash"></i></a></td>
            </tr>
        @endforeach
    @endisset

    <h2>{{ (isset($selectedCategory)) ? 'EDIT CATEGORY' : 'ADD CATEGORY'
    }}</h2> <form action="{{ { (isset($selectedCategory)) ? asset('/admin-
    panel/categories/update/'.' $selectedCategory->category_id) : route('insertCategory')
    }}" method="post">
        {{ csrf_field() }}
        <div class="field half first">
            <label for="categoryNameAdmin">Name</label>
```

```

        <input type="text" name="categoryNameAdmin" id="categoryNameAdmin" value="{{
(isset($selectedCategory)) ? $selectedCategory->category_name : old('categoryNameAdmin')
}}"/>
    </div>
    <div class="form-group">
        <input type="submit" name="addEditCategory" value="{{
(isset($selectedCategory))? 'UPDATE' : 'ADD' }}" class="btn btn-default"/>
        @isset($selectedCategory)
            <a href="{{ { asset('admin-panel/categories/') }} }}" class="button
cancel">CANCEL</a> @endisset
    </div>
</form>
</div>

@endsection

```



## admin/adminNavigation.blade.php

```
@extends('pages.admin')

@section('admin')

    @isset($navigations)
        @if(count($navigations) > 5)
            <div class="wrapper-paging">
                <ul>
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li>
                    <li><a class="paging-this"><strong>0</strong> of <span>x</span></a></li>
                    <li><a class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
                </ul>
            </div>
        @endif
    @endisset

    @isset($navigations)
        @foreach($navigations as $navigation)
            <tr>
                <td>{{ $navigation->navigation_id }}</td>
                <td>{{ $navigation->navigation_name }}</td>
                <td>{{ $navigation->navigation_path }}</td>
                <td><i class="icon {{ $navigation->navigation_icon }}"></i> ({{ $navigation->navigation_icon }})</td>
                <td>{{ date('d.m.Y', strtotime($navigation->created_at)) }}</td>
                <td>{{ (isset($navigation->updated_at)) ? date('d.m.Y', strtotime($navigation->updated_at)) : 'NEVER' }}</td>
                <td><a href="{{ { asset('/admin-panel/navigation/'.$navigation->navigation_id) }}"><i class="fa fa-edit"></i></a></td>
                <td><a href="{{ { asset('/admin-panel/navigation/delete/'.$navigation->navigation_id) }}"><i class="fa fa-trash"></i></a></td>
            </tr>
        @endforeach
    @endisset

    <h2>{{ (isset($selectedNavigation)) ? 'EDIT NAVIGATION LINK' : 'ADD NAVIGATION LINK' }}</h2>
    <form action="{{ (isset($selectedNavigation)) ? asset('/admin-panel/navigation/update/'.$selectedNavigation->navigation_id) : route('insertNavigation') }}"
    method="post">
        {{ csrf_field() }}
```

```

<div class="field half first">
  <label for="navigationNameAdmin">Name</label>
  <input type="text" name="navigationNameAdmin"
id="navigationNameAdmin" value="{{ (isset($selectedNavigation)) ?
$selectedNavigation->navigation_name : old('navigationNameAdmin') }}" />
</div>
<div class="field half">
  <label for="navigationPathAdmin">Path</label>
  <input type="text" name="navigationPathAdmin" id="navigationPathAdmin" value="{{
(isset($selectedNavigation)) ? $selectedNavigation->navigation_path :
old('navigationPathAdmin') }}" />
</div>
<div class="field half first">
  <label for="navigationIconAdmin">Icon (manually choose from <a
href="https://fontawesome.com/icons?d=gallery" target="_blank">Font Awesome</a>
website)</label>
  <input type="text" name="navigationIconAdmin" id="navigationIconAdmin"
value="{{ (isset($selectedNavigation)) ? $selectedNavigation->navigation_icon :
old('navigationIconAdmin') }}" />
</div>
<div class="form-group">
  <input type="submit" name="addEditNavigation" value="{{ {
(isset($selectedNavigation)) ? 'UPDATE' : 'ADD' }}" class="btn btn-default"/>
  @isset($selectedNavigation)
    <a href="{{ { asset('admin-panel/navigation/') }}" class="button
cancel">CANCEL</a> @endisset
  </div>
</form>
</div>

@endsection

```

## admin/adminPicture.blade.php

```
@extends('pages.admin')
```

```
@section('admin')
```

```
    @isset($pictures)
```

```
        @if($pictures->count() > 5)
```

```
            <div class="wrapper-paging">
```

```
                <ul>
```

```
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li>
```

```
                    <li><a class="paging-this"><strong>0</strong> of <span>x</span></a></li>
```

```
                    <li><a class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
```

```
                </ul>
```

```
            </div>
```

```
        @endif
```

```
    @endisset
```

```
    @isset($pictures)
```

```
        @foreach($pictures as $picture)
```

```
            <tr>
```

```
                <td>{{ $picture->picture_id }}</td>
```

```
                <td class="show">PREVIEW</td>
```

```
                <td>{{ $picture->picture_name }}</td>
```

```
                <td>{{ $picture->category_name }}</td>
```

```
                <td>{{ $picture->user_name }}</td>
```

```
                <td>{{ date('d.m.Y', strtotime($picture->shared_at)) }}</td>
```

```
                <td>{{ (isset($picture->pictureUpdated)) ? date('d.m.Y',
strtotime($picture->pictureUpdated)) : 'NEVER' }}</td>
```

```
                <td><a href="{{ asset('/admin-panel/pictures/'.$picture->picture_id)
}}"><i class="fa fa-edit"></i></a></td>
```

```
                <td><a href="{{ asset('/admin-panel/pictures/delete/'.$picture->picture_id)
}}"><i class="fa fa-trash"></i></a></td>
```

```
                <td class="hide"></td>
```

```
            </tr>
```

```
        @endforeach
```

```
    @endisset
```

```
    @isset($pictures)
```

```
        @foreach($pictures as $picture)
```

```
            <tr>
```

```
                <td>{{ $picture->picture_id }}</td>
```

```

        <td class="show">PREVIEW</td>
        <td>{{ $picture->picture_name }}</td>
        <td>{{ $picture->category_name }}</td>
        <td>{{ $picture->user_name }}</td>
        <td>{{ date('d.m.Y', strtotime($picture->shared_at)) }}</td>
        <td>{{ (isset($picture->pictureUpdated)) ? date('d.m.Y',
strtotime($picture->pictureUpdated)) : 'NEVER' }}</td>
        <td><a href="{{ asset('/admin-panel/pictures'.'/'.$picture->picture_id)
}}"><i class="fa fa-edit"></i></a></td>
        <td><a href="{{ asset('/admin-panel/pictures/delete/'.'/'.$picture->picture_id)
}}"><i class="fa fa-trash"></i></a></td>
        <td class="hide"></td>
    </tr>
    @endforeach
@endisset
</tbody>
</table>
</div>
</div>
<div class="social column">
    <h2>{{ (isset($selectedPicture)) ? 'EDIT PICTURE' : 'ADD PICTURE'
}}</h2> <form action="{{ (isset($selectedPicture)) ? asset('/admin-
panel/pictures/update/' . $selectedPicture->picture_id) : route('insertPicture') }}"
method="post" enctype="multipart/form-data">
    {{ csrf_field() }}
    <div class="field half first">
        <label for="pictureNameAdmin">Name</label>
        <input type="text" name="pictureNameAdmin" id="pictureNameAdmin" value="{{ (
isset($selectedPicture)) ? $selectedPicture->picture_name : old('pictureNameAdmin') }}" />
    </div>
    <div class="field half">
        <label for="pictureFileAdmin">Picture</label>
        <input type="file" name="pictureFileAdmin" id="pictureFileAdmin" value="{{ (
isset($selectedPicture)) ? $selectedPicture->picture_path : old('pictureFileAdmin') }}" />
    </div>
    <div class="field half first">
        <label for="pictureCategoryAdmin">Category</label>
        <select name="pictureCategoryAdmin" id="pictureCategoryAdmin">
            <option value="0">Choose</option>
            @isset($categories)
            @foreach($categories as $category)

```

```

        <option value="{ { $category->category_id } }" { { (isset($selectedPicture) &&
$selectedPicture->category_id == $category->category_id) ? 'selected' :
(old('pictureCategoryAdmin')== $category->category_id)? 'selected' : " } }>{ { $category-
>category_name } }</option>
        @endforeach
        @endisset
    </select>
</div>
<div class="field half">
    <label for="pictureUserAdmin">User:</label>
    <select name="pictureUserAdmin" id="pictureUserAdmin">
        <option value="0">Choose</option>
        @isset($users)
            @foreach($users as $user)
                <option value="{ { $user->user_id } }" { { (isset($selectedPicture) &&
$selectedPicture->user_id == $user->user_id) ? 'selected' : (old('pictureUserAdmin')== $user-
>user_id) ? 'selected' : " } }>{ { $user->user_name } }</option>
            @endforeach
            @endisset
        </select>
    </div>
    <div class="form-group">
        <input type="submit" name="addEditPicture" value="{ {
(isset($selectedPicture)) ? 'UPDATE' : 'ADD' } }" class="btn btn-default"/>
        @isset($selectedPicture)
            <a href="{ { asset('admin-panel/pictures/') } }" class="button
cancel">CANCEL</a> @endisset
    </div>
</form>
</div>

@endsection

```

## admin/adminPolls.blade.php

```
@extends('pages.admin')

@section('admin')

    @isset($polls)
        @if(count($polls) > 5)
            <div class="wrapper-paging">
                <ul>
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li> <li><a
                    class="paging-this"><strong>0</strong> of <span>x</span></a></li> <li><a
                    class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
                </ul>
            </div>
        @endif
    @endisset

    @isset($polls)
        @foreach($polls as $poll)
            <tr>
                <td>{{ $poll->poll_id }}</td>
                <td>{{ $poll->poll_question }}</td>
                <td>{{ date('d.m.Y', strtotime($poll->created_at)) }}</td>
                <td>{{ (isset($poll->updated_at)) ? date('d.m.Y', strtotime($poll->updated_at)) :
'NEVER' }}</td>
                <td>@if($poll->poll_active == 1)<p class="activation">ACTIVE</p> @else <a
href="{{ { asset('/admin-panel/polls/activate/' . $poll->poll_id) }}">ACTIVATE</a> @endif</td>
                <td><a href="{{ { asset('/admin-panel/polls/' . $poll->poll_id) }}"><i class="fa
fa-edit"></i></a></td>
                <td><a href="{{ { asset('/admin-panel/polls/delete/' . $poll->poll_id)
}} }}"><i class="fa fa-trash"></i></a></td>
            </tr>
        @endforeach
    @endisset

    <h2>{{ (isset($selectedPoll)) ? 'EDIT POLL' : 'ADD POLL' }}</h2>
    <form action="{{ (isset($selectedPoll)) ? asset('/admin-panel/polls/update/' . $selectedPoll[0]-
>poll_id) : route('insertPoll') }}" method="POST">
        {{ csrf_field() }}
        <div class="question">
            <label for="pollQuestionAdmin">Question</label>
```

```

        <input type="text" name='pollQuestionAdmin' value='{{ (isset($selectedPoll))
? $selectedPoll[0]->poll_question : old('pollQuestionAdmin')}}'>
    </div>
    <div class="field half first">
        <label for="pollFirstAnswer">Answer 1</label>
        <input type="text" name='pollFirstAnswer' value='{{ (isset($selectedPoll))
? $selectedPoll[0]->answer : old('pollFirstAnswer')}}'>
        <input type="text" value="Number of votes: {{ (isset($selectedPoll)) ? $selectedPoll[0]-
>answer_votes : 'NO VOTES'}}" disabled="" />
        <input type="hidden" name="answer" value="{{ (isset($selectedPoll)) ? $selectedPoll[0]-
>answer_id : old('answer')}}" />
    </div>
    <div class="field half">
        <label for="pollSecondAnswer">Answer 2</label>
        <input type="text" name='pollSecondAnswer' value='{{ (isset($selectedPoll))
? $selectedPoll[1]->answer : old('pollSecondAnswer')}}'>
        <input type="text" value="Number of votes: {{ (isset($selectedPoll)) ? $selectedPoll[1]-
>answer_votes : 'NO VOTES'}}" disabled="" />
        <input type="hidden" name="answer_alt" value="{{ (isset($selectedPoll))
? $selectedPoll[1]->answer_id : old('answer_alt')}}" />
    </div>
    <input type="submit" class="btn btn-default" value="{{ (isset($selectedPoll)) ? 'UPDATE' :
'ADD'}}" />
    @isset($selectedPoll)
        <a href="{{ asset('admin-panel/polls/') }}" class="button cancel">CANCEL</a>
    @endisset
</form>
</div>

@endsection

```

## admin/adminRoles.blade.php

```
@extends('pages.admin')

@section('admin')

    @isset($roles)
        @if(count($roles) > 5)
            <div class="wrapper-paging">
                <ul>
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li> <li><a
                    class="paging-this"><strong>0</strong> of <span>x</span></a></li> <li><a
                    class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
                </ul>
            </div>
        @endif
    @endisset

    @isset($roles)
        @foreach($roles as $role)
            <tr>
                <td>{{ $role->role_id }}</td>
                <td>{{ $role->role_name }}</td>
                <td>{{ date('d.m.Y', strtotime($role->created_at)) }}</td>
                <td>{{ (isset($role->updated_at)) ? date('d.m.Y', strtotime($role->updated_at)) :
'NEVER' }}</td>
                <td><a href="{{ { asset('/admin-panel/roles/'.$role->role_id) }}"><i class="fa
fa-edit"></i></a></td>
                <td><a href="{{ { asset('/admin-panel/roles/delete/'.$role->role_id)
}} }}"><i class="fa fa-trash"></i></a></td>
            </tr>
        @endforeach
    @endisset

    <h2>{{ (isset($selectedRole)) ? 'EDIT ROLE' : 'ADD ROLE' }}</h2>
    <form action="{{ (isset($selectedRole)) ? asset('/admin-
panel/roles/update/'.$selectedRole->role_id) : route('insertRole') }}" method="post">
        {{ csrf_field() }}
        <div class="field half first">
            <label for="roleNameAdmin">Name</label>
            <input type="text" name="roleNameAdmin" id="roleNameAdmin" value="{{ {
(isset($selectedRole)) ? $selectedRole->role_name : old('roleNameAdmin') }}" />
        </div>
    </form>
```



```
<div class="form-group">
  <input type="submit" name="addEditRole" value="{ { (isset($selectedRole))? 'UPDATE'
: 'ADD' } }" class="btn btn-default"/>
  @isset($selectedRole)
    <a href="{ { asset('admin-panel/roles/') } }" class="button cancel">CANCEL</a>
  @endisset
</div>
</form>
</div>

@endsection
```

## admin/adminUser.blade.php

```
@extends('pages.admin')

@section('admin')

    @isset($users)
        @if(count($users) > 5)
            <div class="wrapper-paging">
                <ul>
                    <li><a class="paging-back"><i class="fa fa-arrow-left"></i></a></li> <li><a
                    class="paging-this"><strong>0</strong> of <span>x</span></a></li> <li><a
                    class="paging-next"><i class="fa fa-arrow-right"></i></a></li>
                </ul>
            </div>
        @endif
    @endisset

    @isset($users)
        @foreach($users as $user)
            <tr>
                <td>{{ $user->user_id }}</td>
                <td>{{ $user->user_mail }}</td>
                <td>{{ $user->user_name }}</td>
                <td>{{ $user->role_name }}</td>
                <td>{{ date('d.m.Y', strtotime($user->registered_at)) }}</td>
                <td>{{ (isset($user->changed_at)) ? date('d.m.Y', strtotime($user->changed_at)) :
'NEVER' }}</td>
                <td><a href="{{ asset('/admin-panel/users/'.$user->user_id) }}"><i class="fa fa-
edit"></i></a></td>
                <td><a href="{{ asset('/admin-panel/users/delete/'.$user->user_id) }}"><i
class="fa fa-trash"></i></a></td>
            </tr>
        @endforeach
    @endisset

    <h2>{{ (isset($selectedUser)) ? 'EDIT USER' : 'ADD USER' }}</h2>
    <form action="{{ (isset($selectedUser)) ? asset('/admin-panel/users/update/'.$selectedUser-
>user_id) : route('insertUser') }}" method="post">
        {{ csrf_field() }}
        <div class="field half first">
            <label for="userEmailAdmin">E-Mail</label>
```

```

        <input type="email" name="userEmailAdmin" id="userEmailAdmin" value="{ {
(isset($selectedUser)) ? $selectedUser->user_mail : old('userEmailAdmin') } }"/>
    </div>
    <div class="field half">
        <label for="userUsernameAdmin">Username</label>
        <input type="text" name="userUsernameAdmin" id="userUsernameAdmin" value="{ {
(isset($selectedUser)) ? $selectedUser->user_name : old('userUsernameAdmin') } }"/>
    </div>
    <div class="field half">
        <label for="userPasswordAdmin">Password</label>
        <input type="password" name="userPasswordAdmin" id="userPasswordAdmin"
value="{ { (isset($selectedUser)) ? $selectedUser->user_pass : old('userPasswordAdmin') } }"/>
    </div>
    <div class="field half left">
        <label for="userRoleAdmin">Role</label>
        <select name="userRoleAdmin"
id="userRoleAdmin"> <option
value="0">Choose</option> @isset($roles)
    @foreach($roles as $role)
        <option value="{ { $role->role_id } }" { { (isset($selectedUser) && $selectedUser-
>role_id == $role->role_id) ? 'selected' : (old('userRoleAdmin')== $role->role_id) ? 'selected' : "
}}>{ { $role->role_name } }</option>
    @endforeach
    @endisset
</select>
</div>
    <div class="form-group">
        <input type="submit" name="addEditUser" value="{ { (isset($selectedUser)) ? 'UPDATE'
: 'ADD' } }" class="btn btn-default"/>
        @isset($selectedUser)
            <a href="{ { asset('admin-panel/users/') } }" class="button
cancel">CANCEL</a> @endisset
        </div>
</form>
</div>

@endsection

```

## ajax/ajaxGallery.blade.php

```
@isset($cats)
    @if($cats->isEmpty())
        <div class="empty">
            <h1>SORRY, THERE ARE NO PICTURES IN THIS CATEGORY. PLEASE TRY
ANOTHER.</h1>
        </div>
    @else
        @foreach($cats as $cat)
            <div class="media">
                <a href="{{ asset('/pictures/'.$cat->picture_id ) }}">picture_name) }}" /></a>
            </div>
        @endforeach
    @endif
@endisset
```

## routes/web.php

```
<?php

Route::group(['middleware' => 'user'], function()
{
    Route::post('/pictures/{id}/comment', 'CommentController@insertComment');

    Route::post('/pictures/{id}/update/{cId}', 'CommentController@updateComment');

    Route::get('/pictures/{id}/delete/{cId}', 'CommentController@deleteComment');

    Route::get('/share', 'FrontEndController@getShare');

    Route::post('/share/share', 'PictureController@sharePicture')->name('sharePicture');

    Route::get('/share/delete/{id}', 'PictureController@deletePicture');

    Route::post('/share/change/{id}', 'UserController@changeUser');

    Route::get('/logout', 'UserController@logout')->name('logout'); });

Route::group(['middleware' => 'admin'], function()
{
    Route::get('/admin-panel/users/{id?}', 'AdminController@showUsers'); Route::post('/admin-panel/users/insert', 'UserController@insertUser')->name('insertUser'); Route::post('/admin-panel/users/update/{id}', 'UserController@updateUser'); Route::get('/admin-panel/users/delete/{id}', 'UserController@deleteUser'); Route::get('/admin-panel/pictures/{id?}', 'AdminController@showPictures');

    Route::post('/admin-panel/pictures/insert', 'PictureController@insertPicture')->name('insertPicture');

    Route::post('/admin-panel/pictures/update/{id}', 'PictureController@updatePicture');

    Route::get('/admin-panel/pictures/delete/{id}', 'PictureController@deletePicture');

    Route::get('/admin-panel/categories/{id?}', 'AdminController@showCategories');
```

```

Route::post('/admin-panel/categories/insert',
'CategoryController@insertCategory')->name('insertCategory');

Route::post('/admin-panel/categories/update/{id}', 'CategoryController@updateCategory');

Route::get('/admin-panel/categories/delete/{id}', 'CategoryController@deleteCategory');

Route::get('/admin-panel/navigation/{id?}', 'AdminController@showNavigation');

Route::post('/admin-panel/navigation/insert', 'NavigationController@insertNavigation')-
>name('insertNavigation');

Route::post('/admin-panel/navigation/update/{id}',
'NavigationController@updateNavigation');

Route::get('/admin-panel/navigation/delete/{id}', 'NavigationController@deleteNavigation');

Route::get('/admin-panel/roles/{id?}', 'AdminController@showRoles');

Route::post('/admin-panel/roles/insert', 'RoleController@insertRole')->name('insertRole');

Route::post('/admin-panel/roles/update/{id}', 'RoleController@updateRole');

Route::get('/admin-panel/roles/delete/{id}', 'RoleController@deleteRole');

Route::get('/admin-panel/polls/{id?}', 'AdminController@showPolls');

Route::post('/admin-panel/polls/insert', 'PollController@insertPoll')->name('insertPoll');

Route::get('admin-panel/polls/activate/{id}', 'PollController@activatePoll');

Route::post('/admin-panel/polls/update/{id}', 'PollController@updatePoll');

Route::get('/admin-panel/polls/delete/{id}', 'PollController@deletePoll');
});

Route::get('/', 'FrontEndController@getHome');

Route::get('/home', 'FrontEndController@getHome');

Route::post('/home/register', 'UserController@register')->name('register');

```

```
Route::post('/home/login', 'UserController@login')->name('login');

Route::get('/home/showPoll', 'PollController@showPoll');

Route::post('/home/insertVote', 'PollController@insertVote')->name('insertVote');

Route::get('/gallery', 'FrontEndController@getGallery');

Route::get('/gallery/{cat}', 'FrontEndController@showCategorized');

Route::get('/pictures/{id}/{cId?}', 'FrontEndController@getPicture');

Route::get('/information', 'FrontEndController@getInformation');

Route::get('/information/download', 'FrontEndController@download')->name('download');

Route::post('/information/mail', 'MailController@sendMail')->name('mail');
```