



WEB PROGRAMIRANJE PHP2

DOKUMENTACIJA PROJEKTA

Student:

Nemanja Lukic 135/15

Beograd 2018.

SADRŽAJ

1. Opis funkcionalnosti.....	1
2. Skica struktura stranica	1
3. Dijagram baze podataka.....	3
3. MVC Organizacija.....	4
Kontroleri.....	4
Modeli.....	5
View.....	7
4. JavaScript kod.....	9
5. PHP kod	15

1.Opis funkcionalnosti

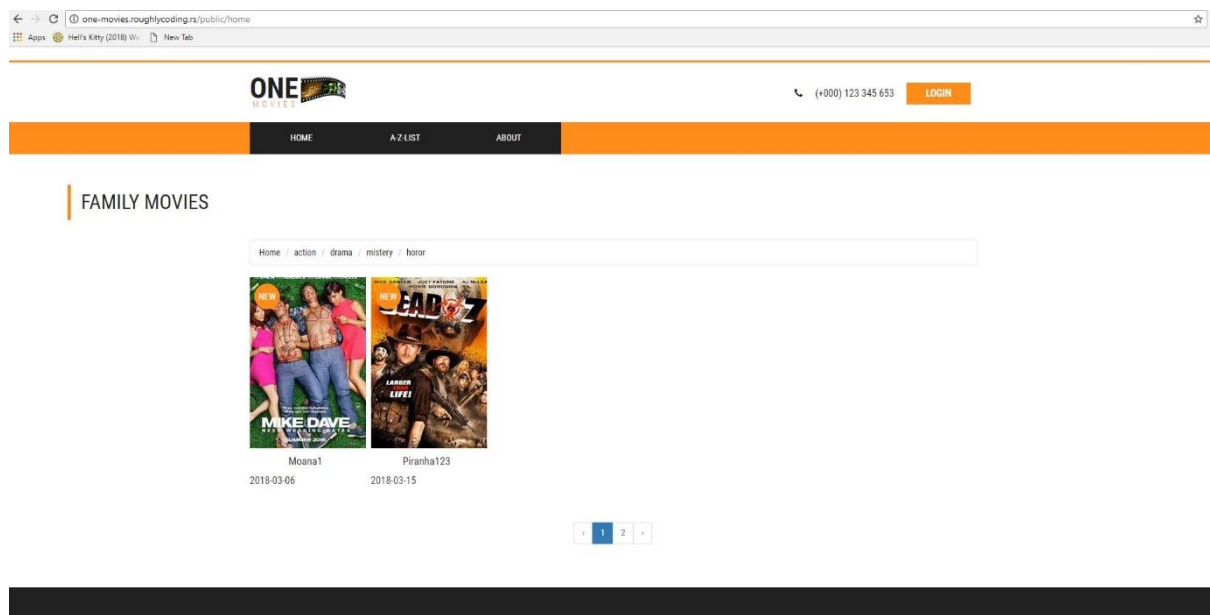
U okviru ove sekcije navedite detaljan opis funkcionalnosti u formi kratke priče ili kreirajte *use case* dijagram.

Korišćeni jezici	<ul style="list-style-type: none">• HTML 4 i HTML 5• CSS• javaScript
Korišćeni HTML šabloni (framework-ci)	Bootstrap
Korišćene biblioteke i njihovi url-ovi odakle ste ih preuzeli	<ul style="list-style-type: none">• jQuery
Korišćene biblioteke iz framework-a	<ul style="list-style-type: none">• Laravel
Korišćene php biblioteke van fremework-a	
Koje funkcionalnosti su realizove putem AJAX-a	Sortiranje po Zanru,Sortiranje po slovima,Anketa
Navešti lokacije gde je napisan AJAX kod (fajlovi ili delovi html stranice)	public/js/skripta.js
Navešti lokacije gde je napisan sopstveni javaScript kod (fajlovi ili delovi html stranice)	Public/js/skripta.js
URL sajta/aplikacije	http://one-movies.roughlycoding.rs
Pristupni parametri	

.

2. Skica struktura stranica

U okviru ove sekcije potrebno je navesti skice osnovnih tipova strana koje se koriste u okviru web sajta/aplikacije. Da bi kreirali skice možete koristiti alatku *Pencil* ili bilo koji drugi alat za kreiranje *wireframe*-a. Takođe moguće je odraditi i *print screen* strukture stranice.



SB Admin v2.0

Search...

Korisnici

Filmovi



Zanrovi

Navigacija

Komentari

Artikla

logout

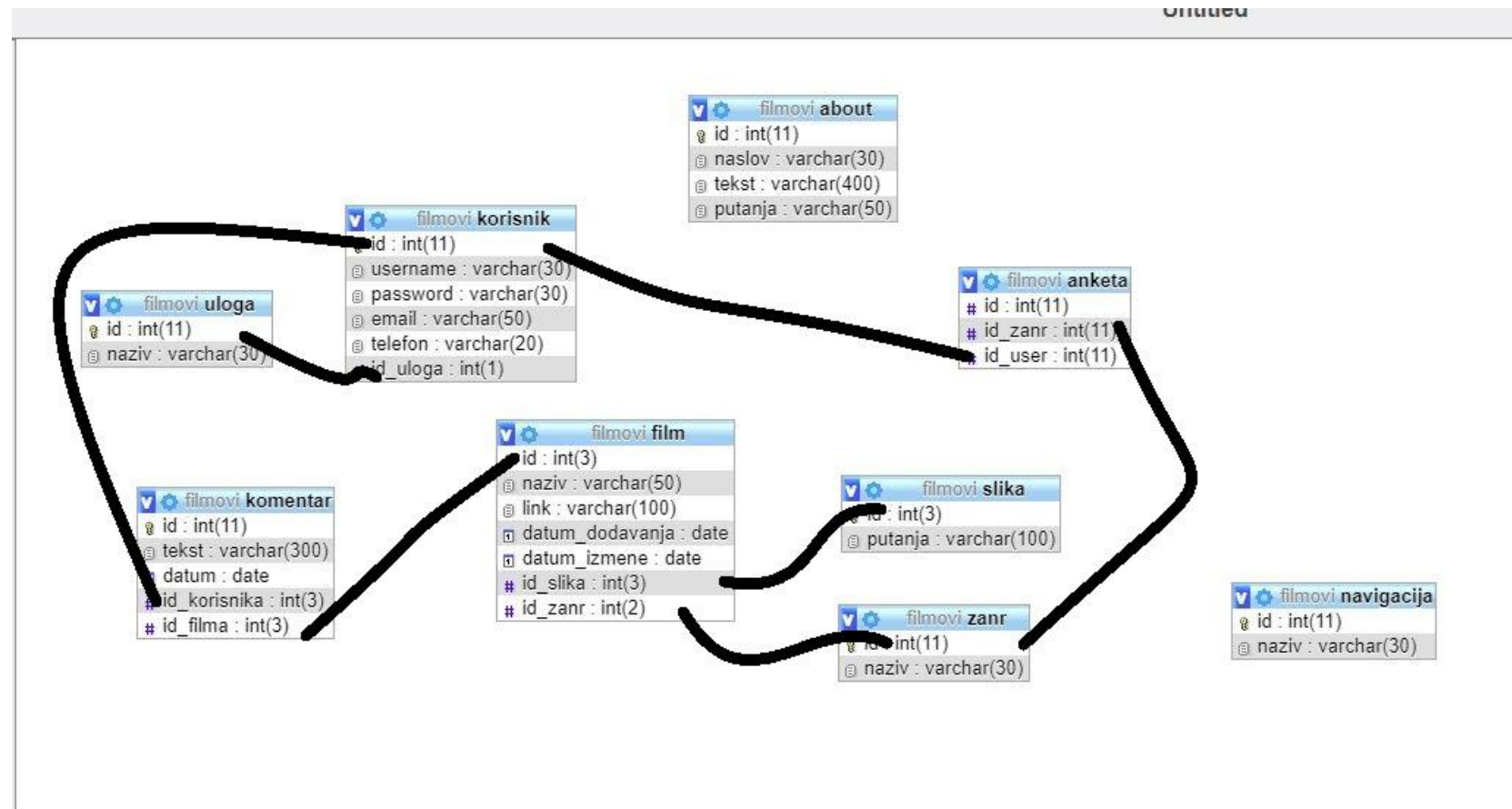
Id	Naziv	Datum_codavanja	Datum_izmene	slika	Zanr
1	Moana1	2018-03-05	2018-03-05		action
4	Holidays	2018-03-17	2018-03-17		action

< 1 2 >

Naziv

Embeded Link

3. Dijagram baze podataka



3.MVC Organizacija

Kontroleri

FrontController
home()
List()
Sortiraj()
About()
Single()
Login()
Register()
Logout()
dodajKomentar()
ObrisiKomentar()
Komentari()
Anketa()

Admin
Korisnici()
Izbrisi_Korisnika()
Dodaj_korisnika()
Korisnici_izmena()
Navigacija()
Dodaj_navigaciju()
Izmeni_navigaciju()
Navigacija_update()
Zarn()
ZanrAdd()
ObrisiKomentar()
zanrDelete()
zanrUpdate()
zanrIzmeni()
Dohvati()
filmAdd()
filmDelete()
filmUpdate()

FilmIzmeni()
Anketa()
anketaDelete()

Jedan kontroler jedna tabela (ovaj tekst ne morate stavljati u dokumentaciju)

Modeli

About
Get()

Filmovi
dohvatiSve()
dohvatiJedan()
Lista()
Sortiraj()
dohvatiZanrove()
sortirajZanr()
dohvatiAdmin()
filmAdd()
dohvatiPutanju
filmDelete()
getOne
filmUpdate()
filmIzmeni2()

Komentar
dohvatisve()

Add()
Delete()
Dohvati()

Korisnik
Login()
Registruj()
Get()
Delete()
Dodaj_korisnika()
dohvatiJednog()
Izmeni_korisnika()

Slika
Add()
izbrisiSliku()

Meni
getAll()

Navigacija
Get()
Delete()
Insert()
getOne()
Update()

Zanr
Get()
getOne()
Delete()
update()
Add()
Slika
dohvatiAnketu()

Obrisiizankete()

Jedan model jedna tabela (ovaj tekst ne morate stavljati u dokumentaciju)

View

components
Footer.blade.php
header.blade.php
Nav.blade.php

layouts
admin.blade.php
front.blade.php

pages
about.blade.php
adminNavigacija.blade.php
adminZanr.blade.php
anketa.blade.php
filmovi.blade.php
filmoviIzmeni.blade.php
home.blade.php
komentari.blade.php
korisnici.blade.php
korisnici_izmena.blade.php
list.blade.php
navigacijaIzmeni.blade.php
panel.blade.php
pocetna.blade.php

single.blade.php
zanrIzmeni.blade.php

4. JavaScript kod

```
$(document).ready(function(){

    $('.sortiranje').click(function(){

        var slovo=$(this).text();

        $.ajax({

            url:'http://localhost/filmovi/public/sortiraj/'+slovo,

            type:"GET",

            dataType:"Text",

            success:function(sortirani){

                var niz=sortirani.slice(1);

                var ispis="";

                var brojac=0;

                niz=jQuery.parseJSON(niz);

                $.each(niz,function(index,film){

                    brojac++

                    ispis+=` <tr>

                        <td>`+brojac+`</td>

                        <td class="w3-list-img"><a href="single/"+film.id_filma+`"> <span>`+film.ime+`</span></a></td>

                        <td>`+film.datum_dodavanja+`</td>

                        <td class="w3-list-info"><a href="genres.html"></a>`+film.naziv+`</td>

                    </tr>`

                });

                $('#lista').html(ispis);
```

```

    },
    error : function(sortirani) {
        //console.log(sortirani.responseText);
    }

});

});

$('.zanr').click(function(){
    var zanr=$(this).text();
    $.ajax({
        type:'GET',
        url:'http://localhost/filmovi/public/sortirajZanr/'+zanr,
        success:function(filmovi){
            var ispis=""
            $.each(filmovi,function(index,film){
                ispis+=`<div class="col-md-2 w3l-movie-gride-agile">
                    <a href="http://localhost/filmovi/public/single/^+film.id_filma+`"
class="hvr-shutter-out-horizontal">
                    <div class="w3l-action-icon"><i class="fa fa-play-circle" aria-
hidden="true"></i></div>
                </a>
                <div class="mid-1">
                    <div class="w3l-movie-text">
                        <h6><a href="single.html">`+film.ime+`</a></h6>
                    </div>
                    <div class="mid-2">

```

```

        <p>`+film.datum_dodavanja+`</p>

        <div class="clearfix"></div>

    </div>

    <div class="ribben two">

        <p>NEW</p>

    </div>

</div>`

    });

    $('#filmovi').html(ispis);

    $('#render').html("");

}

})

});

function registracija(){

    var greske=[];

    var username=document.querySelector('#username').value;

    var password=document.querySelector('#password').value;

    var email=document.querySelector('#email').value;

    var telefon=document.querySelector('#telefon').value;

    var reemail=/^[a-zA-Z0-9.]+@[a-zA-Z0-9](?:[a-zA-Z0-9]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9]{0,61}[a-zA-Z0-9])?)*$/;

    var retelefon=/^[+0-9]{8,15}$/;

    var repassword=/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,}$/;

    var reusername=/^[A-Za-z]{1}[A-Za-z0-9]{5,31}$/;

```

```

if(!reusername.test(username)){

    greske.push("<p class='text-danger'>username nije u dobrom formatu!</p>");

}

if(!repassword.test(password)){

    greske.push("<p class='text-danger'>password nije u dobrom formatu!</p>");

}

if(!retelefon.test(telefon)){

    greske.push("<p class='text-danger'>telefon nije u dobrom formatu!</p>");

}

if(!reemail.test(email)){

    greske.push("<p class='text-danger'>email nije u dobrom formatu!</p>");

}

if(greske.length>0){

    var greska="";

    for(i=0;i<greske.length;i++){

        greska+=greske[i];

        document.querySelector("#greske").innerHTML=greska;

    }

    return false;

}

if(greske.length==0){

    document.querySelector('#greske').innerHTML+=("<p class='text-success'>Uspesno  
ste se Registrovali</p>");

    return true;

```

```
}  
}
```

```
$('#anketaform').on('click','button', function (e) {  
    e.preventDefault();  
    var zanrId=$('#input[name=zanr]:checked').val();  
  
    $.ajax({  
        headers: {  
            'X-CSRF-TOKEN': $('#meta[name="csrf-token"]').attr('content')  
        },  
        type: "post",  
        data:{ zanrid:zanrId},  
        url: 'http://localhost/nemanja/public/anketa',  
        success:function (res) {  
            console.log(res);  
            $respond=res;  
            if($respond == 1){  
                alert('ne mozete dva puta glasati');  
            }  
            else if($respond == 2){  
                alert('Dobar izbor!');  
            }  
            else alert('probajte kasnije');
```



```
    }  
    });  
    });  
    });
```

5. PHP kod

U okviru ove sekcije navesti PHP kod koji je student isključivo sam napisao. Prored uzmedju redova ne sme biti veći od 6pt.

Admin.php

```
<?php
```

```
/**
```

```
 * Created by PhpStorm.
```

```
 * User: korisnik
```

```
 * Date: 15-Mar-18
```

```
 * Time: 22:35
```

```
*/
```

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

```
use App\Models\Anketa;
```

```
use App\Models\Filmovi;
```

```
use App\Models\Korisnik;
```

```
use App\Models\Navigacija;
```

```
use App\Models>Zanr;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\File;
```

```
use App\Models\Slika;
```

```
use Validator;
```

```
class Admin
```

```
{
```

```

private $data=[];

public function admin(){
    return view('pages.panel');
}

public function korisnici()
{
    $korisnik=new Korisnik();
    $this->data['korisnici']=$korisnik->get();
    return view('pages.korisnici',$this->data);
}

public function izbrisi_korisnika($id){
    $korisnik=new Korisnik();
    $rez=$korisnik->delete($id);
    if($rez){
        return redirect()->route('korisnici');
    }
}

public function dodaj_korisnika(Request $request){
    $rules = [
        'username' => 'min:5|max:30',
        'password' => 'min:5|max:30',
        'email' => 'email',
        'telefon' => 'alpha_num'
    ];
    $request->validate($rules);
    $username=$request->get('username');

```

```

$password=$request->get('password');

$email=$request->get('email');

$telefon=$request->get('telefon');

$id_uloa=$request->get('id_uloa');

$korisnik=new Korisnik();

try{

    $upit=$korisnik->dodaj_korisnika($username,$password,$email,$telefon,$id_uloa);

    return redirect()->route('korisnici');

} catch(\Exception $e){

    \Log::error('Problem sa fajlom!! ' . $e->getMessage());

}

}

public function korisnici_izmena($id){

    $korisnik=new Korisnik();

    $this->data['korisnik']=$korisnik->dohvatiJednog($id);

    return view('pages.korisnici_izmena',$this->data);

}

public function izmeni_korisnika(Request $request){

    $rules = [

        'username' => 'min:5|max:30',

        'password' => 'min:5|max:30',

        'email' => 'email',

        'telefon' => 'alpha_num'

    ];

    $request->validate($rules);

    $id=$request->get('id');

```

```

$username=$request->get('username');
$password=$request->get('password');
$email=$request->get('email');
$telefon=$request->get('telefon');
$id_uloga=$request->get('id_uloga');
try{ $korisnik=new Korisnik();
$upit=$korisnik->izmeni_korisnika($id,$username,$password,$email,$telefon,$id_uloga);
    return redirect()->route('korisnici');
}catch(\Exception $e){
    \Log::error('Problem sa fajlom!! ' . $e->getMessage());
}

}

public function navigacija(){
    $navigacija=new Navigacija();
    $this->data['navigacija']=$navigacija->get();
    return view('pages.adminNavigacija',$this->data);
}

public function izbrisi_navigaciju($id){
    $navigacija=new Navigacija();
    $upit=$navigacija->delete($id);
    if($upit){
        return redirect()->route('navigacija');
    }
}

public function dodaj_navigaciju(Request $request){
    $rules = [

```

```

        'naziv' => 'required'
    ];

    $request->validate($rules);

    $naziv=$request->get('naziv');

    $navigacija=new Navigacija();

    try{ $upit=$navigacija->insert($naziv);

        return redirect()->route('navigacija');

    }

    catch(\Exception $e){

        \Log::error('Problem sa fajlom!! ' . $e->getMessage());

    }

}

public function izmeni_navigaciju($id){

    $navigacija=new Navigacija();

    $this->data['nav']=$navigacija->getOne($id);

    return view('pages.navigacijaIzmeni',$this->data);

}

public function navigacija_update(Request $request){

    $rules = [

        'naziv' => 'required'

    ];

    $request->validate($rules);

    $id=$request->get('id');

    $naziv=$request->get('naziv');

```

```

$navigacija=new Navigacija();

try{ $upit=$navigacija->update($id,$naziv);

    return redirect()->route('navigacija');

} catch(\Exception $e){

    \Log::error('Problem sa fajlom!! ' . $e->getMessage());

}

}

public function zanr(){

    $zanr=new Zanr();

    $this->data['zanr']=$zanr->get();

    return view('pages.adminZanr',$this->data);

}

public function zanrAdd(Request $request){

    $rules = [

        'naziv' => 'required'

    ];

    $request->validate($rules);

    $naziv=$request->get('naziv');

    $zanr=new Zanr();

    try{

        $upit=$zanr->add($naziv);

        if($upit){

            return redirect()->route('zanr');

        }

    }

    catch(\Exception $e){

        \Log::error('Problem sa fajlom!! ' . $e->getMessage());
    }
}

```

```

    }
}

public function zanrDelete($id){
    $zanr=new Zanr();
    $upit=$zanr->delete($id);
    if($upit){
        return redirect()->route('zanr');
    }
}

public function zanrUpdate($id){
    $zanr=new Zanr();
    $this->data['zanr_one']=$zanr->getOne($id);
    return view('pages.zanrIzmeni',$this->data);
}

public function zanrIzmeni(Request $request){
    $rules = [
        'naziv' => 'required',
    ];
    $request->validate($rules);
    $id=$request->get('id');
    $naziv=$request->get('naziv');
    $zanr=new Zanr();
    try{
        $upit=$zanr->update($id,$naziv);
        return redirect()->route('zanr');
    }catch(\Exception $e){
        \Log::error('Problem sa fajlom!! ' . $e->getMessage());
    }
}

```



```

    }
}

public function dohvati(){
    $filmovi=new Filmovi();

    $this->data['zanrovi']=$filmovi->dohvatiZanrove();

    $this->data['filmovi']=$filmovi->dohvatiAdmin(2);

    return view('pages.filmovi',$this->data);
}

public function filmAdd(Request $request)
{
    $rules = [
        'naziv' => 'required',
        'link' => 'required',
        'slika' => 'required|mimes:jpg,jpeg,png,gif',
        'id_zanr' => 'required'
    ];

    $request->validate($rules);

    $slika = $request->file('slika');

    $tmp_path = $slika->getPathName();

    $extension = $slika->getClientOriginalExtension();

    $folder = 'images/';

    $file_name = time() . "." . $extension;

    $new_path = public_path($folder) . $file_name;

    try {

        // 4 - Upload slike na server

        File::move($tmp_path, $new_path);
    }
}

```

```
// 5 - Unos slike u bazu
```

```
$slika = new Slika();  
$id = $slika->add($file_name);  
if($id){  
    $naziv=$request->get('naziv');  
    $link=$request->get('link');  
    $id_slika=$id;  
    $iz_zanr=$request->get('id_zanr');  
    $film=new Filmovi();  
    $upit=$film->filmAdd($naziv,$link,$id_slika,$iz_zanr);  
    if($upit){  
        return redirect()->route('filmovi');  
    }  
}
```

```
} catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) { //  
greske sa fajlom
```

```
    \Log::error('Problem sa fajlom!! ' . $ex->getMessage());  
    return redirect()->back()->with('error', 'Greska pri dodavanju slike!');  
}
```

```
catch(\Illuminate\Database\QueryException $ex){  
    \Log::error($ex->getMessage());  
    return redirect()->back()->with('error','Greska pri dodavanju posta u bazu!');  
}
```

```
}
```

```
public function filmDelete($id){
```

```

$film=new Filmovi();

$slika=$film->dohvatiPutanju($id);

$putanja=$slika->putanja;

$image_path = public_path("images/$putanja");

File::delete($image_path);

$upit=$film->filmDelete($id);

if($upit){

    return redirect()->route('filmovi');

}

else{

    echo "nista";

}

}

public function filmUpdate($id){

    $zanr=new Zanr();

    $film=new Filmovi();

    $this->data['film']=$film->getOne($id);

    $this->data['zanrovi']=$zanr->get();

    return view('pages.filmoviIzmeni',$this->data);

}

public function filmIzmeni(Request $request)

{

    $slika = $request->file('slika');

    if ($slika != "") {

        $tmp_path = $slika->getPathName();

        $extension = $slika->getClientOriginalExtension();

        $trenutna = $request->get('trenutna');

```

```

$image_path = public_path("images/$trenutna");

File::delete($image_path);

$folder = 'images/';

$file_name = time() . "." . $extension;

$new_path = public_path($folder) . $file_name;

// 4 - Upload slike na server


File::move($tmp_path, $new_path);

// 5 - Unos slike u bazu


$slika = new Slika();

$izbrisi=$slika->izbrisiSliku($trenutna);

if($izbrisi) {

    $id = $slika->add($file_name);

    if ($id) {

        $id_filma = $request->get('id_filma');

        $naziv = $request->get('naziv');

        $link = $request->get('link');

        $id_slika = $id;

        $id_zanr = $request->get('id_zanr');

        $film = new Filmovi();

        $upit = $film->filmUpdate($id_filma, $naziv, $link, $id_slika, $id_zanr);

        if ($upit) {

            return redirect()->route('filmovi');

        }

    }

}

}

```

```

    }else{

        $id_filma = $request->get('id_filma');

        $naziv = $request->get('naziv');

        $link = $request->get('link');

        $id_zanr = $request->get('id_zanr');

        $film = new Filmovi();

        $upit = $film->filmIzmeni2($id_filma, $naziv, $link,$id_zanr);

        if($upit){

            return redirect()->route('filmovi');

        }

    }

}

public function anketa(){

    $anketa=new Anketa();

    $podaci=$anketa->dohvatiAnketu();

    return view('pages.anketa', compact('podaci'));

}

public function anketadelete($id){

    $anketa=new Anketa();

    $anketa->obrisiizankete($id);

    return back();

}

}

```

FrontController.php

```
<?php

/**
 * Created by PhpStorm.
 * User: Nex
 * Date: 26.2.2018.
 * Time: 23.19
 */

namespace App\Http\Controllers;

use App\Models\Meni;
use App\Models\Filmovi;
use App\Models\Korisnik;
use App\Models>About;
use App\Models\Komentar;
use App\Models>Anketa;
use App\Models>Zanr;
use Illuminate\Http\Request;

class FrontController
{
    private $data;

    public function __construct()
    {
```

```

$meni=new Meni();

$this->data['meni'] = $meni->getAll();

$about=new About();

$this->data['about']=$about->get();

}

public function home(){

    $film=new Filmovi();

    $zanr=new Filmovi();

    $this->data['zanrovi']=$zanr->dohvatiZanrove();

    $this->data['filmovi']=$film->dohvatiSve(2);

    return view('pages.home',$this->data);

}

public function list(){

    return view('pages.list',$this->data);

}

public function sortiraj($slovo){

    echo $slovo;

    $sortirani=new Filmovi();

    $niz=$sortirani->sortiraj($slovo);

    return $niz;

}

public function about(){

    $zanr=new Zanr();

    $zanrovi=$zanr->get();


    return view('pages.about',$this->data, compact('zanrovi'));

```

```

}

public function single($id){

    $komentar=new Komentar();

    $this->data['komentar']=$komentar->dohvatisve($id);

    $film=new Filmovi();

    $this->data['jedan']=$film->dohvatiJedan($id);

    return view('pages.single',$this->data);

}

public function sortirajZanr($zanr){

    $filmovi=new Filmovi();

    $upit=$filmovi->sortirajZanr($zanr);

    return $upit;

}

public function login(Request $request){

    $username=$request->get('username');

    $password=$request->get('password');

    $korisnik=new Korisnik();

    $upit=$korisnik->login($username,$password);

    if($upit){

        session()->put('user',$upit);

        return redirect()->route('HOME');

    }

    else{

        return redirect()->route('HOME');

    }

}

```



```

public function register(Request $request){

    $rules = [

        'username' => 'min:5|max:30',

        'password' => 'min:5|max:30',

        'email' => 'email',

        'telefon' => 'alpha_num'

    ];

    $request->validate($rules);

    $username=$request->get('username');

    $password=$request->get('password');

    $email=$request->get('email');

    $telefon=$request->get('telefon');

    $korisnik=new Korisnik();

    try {

        $upit = $korisnik->registruj($username, $password, $email, $telefon);

        return redirect()->back();

    }catch(\Exception $e){

        \Log::error('Problem sa fajlom!! ' . $e->getMessage());

    }

}

public function logout(){

    session()->flush();

    return redirect()->route('HOME');

}

public function dodajKomentar(Request $request){

    $id_korisnik=$request->get('id_korisnik');

```

```

        $id_film=$request->get('id_film');

        $tekst=$request->get('tekst');

        $komentar=new Komentar();

        $upit=$komentar->add($tekst,$id_film,$id_korisnik);

        if($upit){

            return redirect()->route('single',[$id_film]);

        }

    }

    public function obrisiKomentar($id){

        $komentar=new Komentar();

        $upit=$komentar->delete($id);

        if($upit){

            return redirect()->back();

        }

    }

    public function komentari(){

        $komentar=new Komentar();

        $this->data['komentari_admin']=$komentar->dohvati();

        return view('pages.komentari',$this->data);

    }

    public function anketa(Request $request){

        $userid= $request->session()->get('user')->id;

        $zanrid=$request->zanrid;

        $provera=\DB::table('anketa')->where('id_user',$userid)->get()->count();
    
```

```

if($provera == 1){
    return 1;
}
else{
    try{
        \DB::table('anketa')->insert(
            ['id_zanr' => $zanrid,
            'id_user'=> $userid
            ]);

        return 2;
    }catch (\Exception $ex){
        \Log::error('greska prilikom koriscenja ankete' . $ex.getMessage());
        return 0;
    }
}
}

```

About.php

```
<?php
```

```
/**
```

```
* Created by PhpStorm.
```

```
* User: Nex
```

```
* Date: 27.2.2018.
```

```
* Time: 17.29
```

```
*/
```

```
namespace App\Models;
```

```
use Illuminate\Support\Facades\DB;
```

```
class About
```

```
{
```

```
    public function get(){
```

```
        return DB::table('about')->get()->first();
```

```
    }
```

```
}
```

Filmovi.php

```
<?php
```

```
/**
```

```
 * Created by PhpStorm.
```

```
 * User: korisnik
```

```
 * Date: 15-Mar-18
```

```
 * Time: 18:06
```

```
*/
```

```
namespace App\Models;
```

```
class Filmovi
```

```
{
```

```

public function dohvatiSve($broj){
    $upit=\DB::table('film')
        ->select('*', 'slika.id as id_slika')
        ->select('*', 'film.id as id')
        ->join('slika', 'slika.id', '=', 'film.id_slika')
        ->paginate($broj);
    return $upit;
}

public function dohvatiJedan($id){
    $upit=\DB::table('film')
        ->select('*', 'slika.id as id_slika')
        ->join('slika', 'slika.id', '=', 'film.id_slika')
        ->where('film.id', '=', $id)
        ->get()->first();
    return $upit;
}

public function lista($naziv='P'){
    $upit=\DB::table('film')
        ->join('slika', 'slika.id', '=', 'film.id_slika')
        ->where('naziv', 'like', $naziv.'%')
        ->get();
    return $upit;
}

public function sortiraj($slovo){
    $upit=\DB::table('film')
        ->select('*', 'film.naziv as ime', 'film.id as id_filma')
        ->join('slika', 'slika.id', '=', 'film.id_slika')

```

```

->join('zanr','zanr.id','=','film.id_zanr')

->where('film.naziv','like', $slovo.'%')

->get();

return $upit;
}

public function dohvatiZanrove(){
    return \DB::table('zanr')->get();
}

public function sortirajZanr($zanr){
    $upit=\DB::table('film')

    ->select('*', 'film.naziv as ime', 'film.id as id_filma')

    ->join('slika','slika.id','=','film.id_slika')

    ->join('zanr','zanr.id','=','film.id_zanr')

    ->where(['zanr.naziv','=',$zanr]))

    ->get();

    return $upit;
}

public function dohvatiAdmin($broj){
    $upit=\DB::table('film')

    ->select('*', 'film.naziv as ime', 'film.id as id_filma')

    ->join('slika','slika.id','=','film.id_slika')

    ->join('zanr','zanr.id','=','film.id_zanr')

    ->paginate($broj);

    return $upit;
}

public function filmAdd($naziv,$link,$id_slika,$id_zanr){

```

```

$upit=\DB::table('film')

->insert(['naziv' => $naziv,'link' => $link,'id_slika' => $id_slika,'id_zanr' =>
$id_zanr,'datum_dodavanja' => NOW(),'datum_izmene' => NOW()]);

return $upit;
}

public function dohvatiPutanju($id){

$upit=\DB::table('film')

->join('slika','slika.id','=', 'film.id_slika')

->where(['film.id','=', $id])

->get()->first();

return $upit;

}

public function filmDelete($id){

$upit=\DB::table('film')

->where(['id','=', $id])

->delete();

return $upit;

}

public function getOne($id){

$upit=\DB::table('film')

->select('*', 'film.naziv as ime', 'film.id as id_filma')

->join('slika','slika.id','=', 'film.id_slika')

->join('zanr','zanr.id','=', 'film.id_zanr')

->where(['film.id','=', $id])

->get()

->first();

```

```

        return $upit;

    }

    public function filmUpdate($id_filma,$naziv,$link,$id_slika,$id_zanr){

        $upit=\DB::table('film')

            ->where(['film.id','=',$id_filma])

            ->update([

                'naziv' => $naziv,

                'link' => $link,

                'id_slika' => $id_slika,

                'id_zanr' => $id_zanr

            ]);

        return $upit;

    }

    public function filmIzmeni2($id_filma,$naziv,$link,$id_zanr){

        $upit=\DB::table('film')

            ->where(['film.id','=',$id_filma])

            ->update([

                'naziv' => $naziv,

                'link' => $link,

                'id_zanr' => $id_zanr

            ]);

        return $upit;

    }

}

```

Komentar.php


```

<?php

/**
 * Created by PhpStorm.
 * User: korisnik
 * Date: 17-Mar-18
 * Time: 10:39
 */

namespace App\Models;

class Komentar
{
    public function dohvatisve($id){
        $upit=\DB::table('komentar')
            ->select('*', 'komentar.id as id_komentar')
            ->join('film', 'film.id', '=', 'komentar.id_filma')
            ->join('korisnik', 'korisnik.id', '=', 'komentar.id_korisnika')
            ->where(['film.id', '=', $id])
            ->get();

        return $upit;
    }

    public function add($tekst, $id_film, $id_korisnik){
        $upit=\DB::table('komentar')
            ->insert([
                'tekst' => $tekst,
                'id_korisnika' => $id_korisnik,
            ])
    }
}

```

```

        'id_filma' => $id_film,

        'datum' => NOW()

    });

    return $upit;
}

public function delete($id){

    $upit=\DB::table('komentar')

        ->where([[ 'id','=',$id]])

        ->delete();

    return $upit;
}

public function dohvati(){

    $upit=\DB::table('komentar')

        ->select('*', 'komentar.id as id_komentar')

        ->join('film', 'film.id','=', 'komentar.id_filma')

        ->join('korisnik', 'korisnik.id','=', 'komentar.id_korisnika')

        ->get();

    return $upit;
}

}

```

Korisnik.php

```

<?php

/**
 * Created by PhpStorm.
 * User: korisnik
 * Date: 15-Mar-18
 * Time: 21:34

```

```
*/
```

```
namespace App\Models;
```

```
class Korisnik
```

```
{
```

```
    protected $table="korisnik"; //veza sa tabelom korisnik
```

```
    public function login($username,$password){
```

```
        $upit=\DB::table('korisnik')
```

```
            ->join('uloga','uloga.id','=', 'korisnik.id_uloga')
```

```
            ->where([[ 'username' ,'=',$username],[ 'password' ,'=',$password]])
```

```
            ->get()
```

```
            ->first();
```

```
        return $upit;
```

```
    }
```

```
    public function registruj($username,$password,$email,$telefon){
```

```
        $upit=\DB::table('korisnik')
```

```
            ->insert(['username' => $username,'password' => $password,'email' => $email,'telefon'  
=> $telefon,'id_uloga' => 1]);
```

```
        return $upit;
```

```
    }
```

```
    public function get(){
```

```
        $upit=\DB::table('korisnik')
```

```
            ->select('*', 'korisnik.id as id_korisnik')
```

```
            ->join('uloga','uloga.id','=', 'korisnik.id_uloga')
```

```

        ->get();
    return $upit;
}

public function delete($id){
    $upit=\DB::table('korisnik')
        ->where(['id','=', $id])
        ->delete();
    return $upit;
}

public function dodaj_korisnika($username,$password,$email,$telefon,$id_uloga){
    $upit=\DB::table('korisnik')
        ->insert(['username' => $username,'password' => $password,'email' => $email,'telefon'
=> $telefon,'id_uloga' => $id_uloga]);
    return $upit;
}

public function dohvatiJednog($id){
    $upit=\DB::table('korisnik')
        ->select('*', 'korisnik.id as id_korisnik')
        ->join('uloga','uloga.id','=', 'korisnik.id_uloga')
        ->where(['korisnik.id','=', $id])->get()->first();
    return $upit;
}

public function izmeni_korisnika($id,$username,$password,$email,$telefon,$id_uloga){
    $upit=\DB::table('korisnik')
        ->where(['id','=', $id])
        ->update([
            'username' => $username,

```

```

        'password' => $password,
        'email' => $email,
        'telefon' => $telefon,
        'id_uloga' => $id_uloga
    );
    return $upit;
}
}

```

Meni.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Nex
 * Date: 27.2.2018.
 * Time: 01.32
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Meni
{
    public function getAll(){
        $rezultat= DB ::table('navigacija')->get();
        return $rezultat;
    }
}

```

```
}  
}
```

Navigacija.php

```
<?php
```

```
/**
```

```
 * Created by PhpStorm.
```

```
 * User: korisnik
```

```
 * Date: 17-Mar-18
```

```
 * Time: 20:24
```

```
 */
```

```
namespace App\Models;
```

```
class Navigacija
```

```
{
```

```
    public function get(){
```

```
        $upit=\DB::table('navigacija')->get();
```

```
        return $upit;
```

```
    }
```

```
    public function delete($id){
```

```
        $upit=\DB::table('navigacija')
```

```
            ->where(['id','=',$id])
```

```
            ->delete();
```

```
        return $upit;
```

```
    }
```

```
    public function insert($naziv){
```

```

        $upit=\DB::table('navigacija')
            ->insert(['naziv' => $naziv]);

        return $upit;
    }

    public function getOne($id){
        $upit=\DB::table('navigacija')
            ->where(['id','=',$id])
            ->get()
            ->first();

        return $upit;
    }

    public function update($id,$naziv){
        $upit=\DB::table('navigacija')
            ->where(['id','=',$id])
            ->update(['naziv' => $naziv]);

        return $upit;
    }

}

```

Slika.php

```

<?php

/**
 * Created by PhpStorm.
 * User: korisnik
 * Date: 17-Mar-18
 * Time: 22:35
 */

```

```

namespace App\Models;

class Slika
{
    public function add($putanja){
        $upit=\DB::table('slika')
            ->insertGetId(['putanja' => $putanja]);
        return $upit;
    }
    public function izbrisiSliku($putanja){
        $upit=\DB::table('slika')
            ->where(['putanja','=',$putanja])
            ->delete();
        return $upit;
    }
}

```

Zanr.php

```

<?php

/**
 * Created by PhpStorm.
 * User: korisnik
 * Date: 17-Mar-18
 * Time: 20:55
 */

```



```

namespace App\Models;

class Zanr
{

    public function get(){
        $upit=\DB::table('zanr')
            ->get();
        return $upit;
    }

    public function getOne($id){
        $upit=\DB::table('zanr')
            ->where(['id','=',$id])
            ->get();
        return $upit;
    }

    public function delete($id){
        $upit=\DB::table('zanr')
            ->where(['id','=',$id])
            ->delete();
        return $upit;
    }

    public function update($id,$naziv){
        $upit=\DB::table('zanr')
            ->where(['id','=',$id])

```

```

        ->update(['naziv' => $naziv]);

    return $upit;

}

public function add($naziv){

    $upit=\DB::table('zanr')

        ->insert(['naziv' => $naziv]);

    return $upit;

}

}

```

Anketa.php

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Anketa extends Model
```

```

{

    public function dohvatiAnketu(){

        $upit=\DB::table('anketa')

            ->select('*', 'anketa.id as id_anketa')

            ->join('korisnik', 'korisnik.id', '=', 'anketa.id_user')

            ->join('zanr', 'zanr.id', '=', 'anketa.id_zanr')

            ->get();

        return $upit;
    }
}

```

```
}

public function obrisiizankete($id){
    $upit=\DB::table('anketa')->where('id',$id)->delete();
    return $upit;
}
}
```