

VISOKA ŠKOLA STRUKOVNIH STUDIJA
ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

“MovieBlackout bioskop”
WEB SAJT U LARAVEL FRAMEWORKU
Z a v r š n i r a d

Mentor:

Dr Nenad Kojić, dipl. inž.

Student:

Valerija Končar 56/17

Beograd, 2021.

VISOKA ŠKOLA STRUKOVNIH STUDIJA
ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

Predmet: Web programiranje PHP2

Tema: **“MovieBlackout” bioskop**

Ocena ____ (_____)

Članovi komisije:

Sadržaj

1.	UVOD	6
2.	RADNO OKRUŽENJE	7
3.	ORGANIZACIJA RADA	8
3.1	Organizaciona šema	8
3.2	Stranice	9
3.2.1	Home	9
3.2.2	Movie Details	11
3.2.3	Cooming Soon	12
3.2.4	Membership	12
3.2.5	About Cinema	13
3.2.6	About Author	14
3.2.7	Contact Cinema	14
3.2.8	Contact Admin	15
3.2.9	Reservation	15
3.2.10	Regain Account Access	16
3.2.11	Admin	16
4.	KOD	19
4.1	Views	19
4.1.1	home.blade.php	19
4.1.2	movie_details.blade.php	21
4.1.3	coming_soon.blade.php	24
4.1.4	Membership.blade.php	25
4.1.5	about_cinema.blade.php	27
4.1.6	about_author.blade.php	28
4.1.7	contact_cinema.blade.php	28
4.1.8	contact_admin.blade.php	29
4.1.9	reservation.blade.php	30
4.1.10	regain_account_access.blade.php	31
4.1.11	admin.blade.php	32
4.1.12	head.blade.php	39
4.1.13	header.blade.php	39
4.1.14	footer.blade.php	44
4.1.15	comment.blade.php	45
4.1.16	footer_section.blade.php	46
4.1.17	menu_children.blade.php	46
4.1.18	movie.blade.php	47
4.1.19	movie_starts.blade.php	48
4.1.20	projection.blade.php	49
4.1.21	template.blade.php	49
4.1.22	admin_template.blade.php	49
4.1.23	head.blade.php	49
4.1.24	header.blade.php	50
4.1.25	footer.blade.php	60

4.1.26	movie.blade.php	61
4.2	Controllers	62
4.2.1	AdminController.php	62
4.2.2	CommentController.php	63
4.2.3	FrontController.php	65
4.2.4	HomeController.php	66
4.2.5	MemberController.php	67
4.2.6	MenuLinkController.php	68
4.2.7	MessageController.php	69
4.2.8	MovieController.php	70
4.2.9	PHPMailerController.php	72
4.2.10	ProjectionController.php	74
4.2.11	ReservationController.php	75
4.2.12	UserController.php	76
4.2.13	ActivityController.php	78
4.2.14	MovieController.php	79
4.2.15	ProjectionController.php	83
4.3	Models	86
4.3.1	AboutCinema.php	86
4.3.2	Activity.php	87
4.3.3	Actor.php	87
4.3.4	Author.php	88
4.3.5	Comment.php	88
4.3.6	Country.php	90
4.3.7	Director.php	90
4.3.8	Distributer.php	91
4.3.9	Gender.php	91
4.3.10	Genre.php	91
4.3.11	Hall.php	92
4.3.12	Membership.php	92
4.3.13	MenuModel.php	93
4.3.14	Message.php	95
4.3.15	Movie.php	95
4.3.16	PageSectionModel.php	105
4.3.17	Price	105
4.3.18	Projection.php	105
4.3.19	Reservation.php	107
4.3.20	Role.php	109
4.3.21	Technology.php	109
4.3.22	User.php	110
4.4	Requests	112
4.4.1	InsertMovieRequest.php	112
4.4.2	InsertProjectionsRequest.php	113
4.4.3	LoginRequest.php	113
4.4.4	MembershipRequest.php	114
4.4.5	RegistrationRequest.php	115
4.4.6	UpdateMovieRequest.php	116

4.4.7	UpdateProjectionRequest.php	117
4.5	Middleware	117
4.5.1	IsAdminLoggedIn	117
4.5.2	IsLoggedIn.php	118
4.6	JS	118
4.6.1	main.js	118
4.7	CSS	173
4.7.1	Style.css.....	173
5.	Baza.....	238
6.	Zaključak.....	240
7.	Literatura.....	241

1. UVOD

“MovieBlackout” je Web sajt za bioskop, rađen u Laravel-u.

Neautorizovan korisnik će imati mogućnost pregleda repetera filmova, filmova koji treba tek da stignu u bioskope I prijave za člansku karticu bioskopa, kontaktiranje bioskopa i Web sajt admina.

Filmovi se mogu pretraživati po njihovom nazivu, filtrirati prema datumu prikaza projekcija, tehnologijama projekcija(2D, 3D), žanru i sortirati po nazivu u rastućem ili opadajućem redosledu.

Nakon submit-ovanja forme za prijavu za člansku kartu, na unetu email adresu će biti poslat verifikacioni kod koji korisnik potom unosi kako bi se utvrdila ispravnost unetog email-a . Ukoliko je prijava prošla uspešno, korisnik će dobiti povratnu poruku koja to potvrđuje.

Ukoliko korisnik poseduje nalog na našem sajtu, ali je iz nekog razloga zaboravio šifru za isti, moći će da je resetuje, unošenjem svoje email adrese(na koju će biti poslat verifikacioni kod) nakon čega će biti redirektovan na drugu stranici gde će morati da unese novu šifru I dobijeni kod.

Autorizovani korisnik će moći isto što i neautorizovani i da rezerviše sedišta za projekciju, da ocenu filmu, ostavi komentar ispod filma, odgovori na nečiji komentar, menja I briše svoje komentare.

Nakon izbora filma I projekcije(stavka u meniju) ili klikom na aktivnu projekciju, korisnik će biti poslat na stranicu za rezervaciju gde će birati broj karata i sedišta, nakon čega se iznos rezervacije updatuje. Cena karte zavisi od dana u nedelji i tehnologije koju koristi izabrana projekcija.

Admin Web sajta će moći da dodaje nove filmove, menja i briše postojeće, menja podatke korisnika i briše ih, briše neželjene komentare, dodaje, menja i briše meni linkove. Takođe će imati pregled aktivnosti svih admina koje će moći sortirati po datumu.

2. RADNO OKRUŽENJE

Tehnologije:

Back-end:

- PHP (Laravel 8.20.1 – open source PHP framework koji koristi model-view-controller(MVC) arhitekturu. Ovakva arhitektura omogućava dobru separaciju između slojeva aplikacije, što doprinosi boljoj organizaciji koda i lakšem razvoju aplikacije.)

Front-end:

- JavaScript
- jQuery
- HTML
- CSS

Baza podataka:

- MySQL

Razvojni alati:

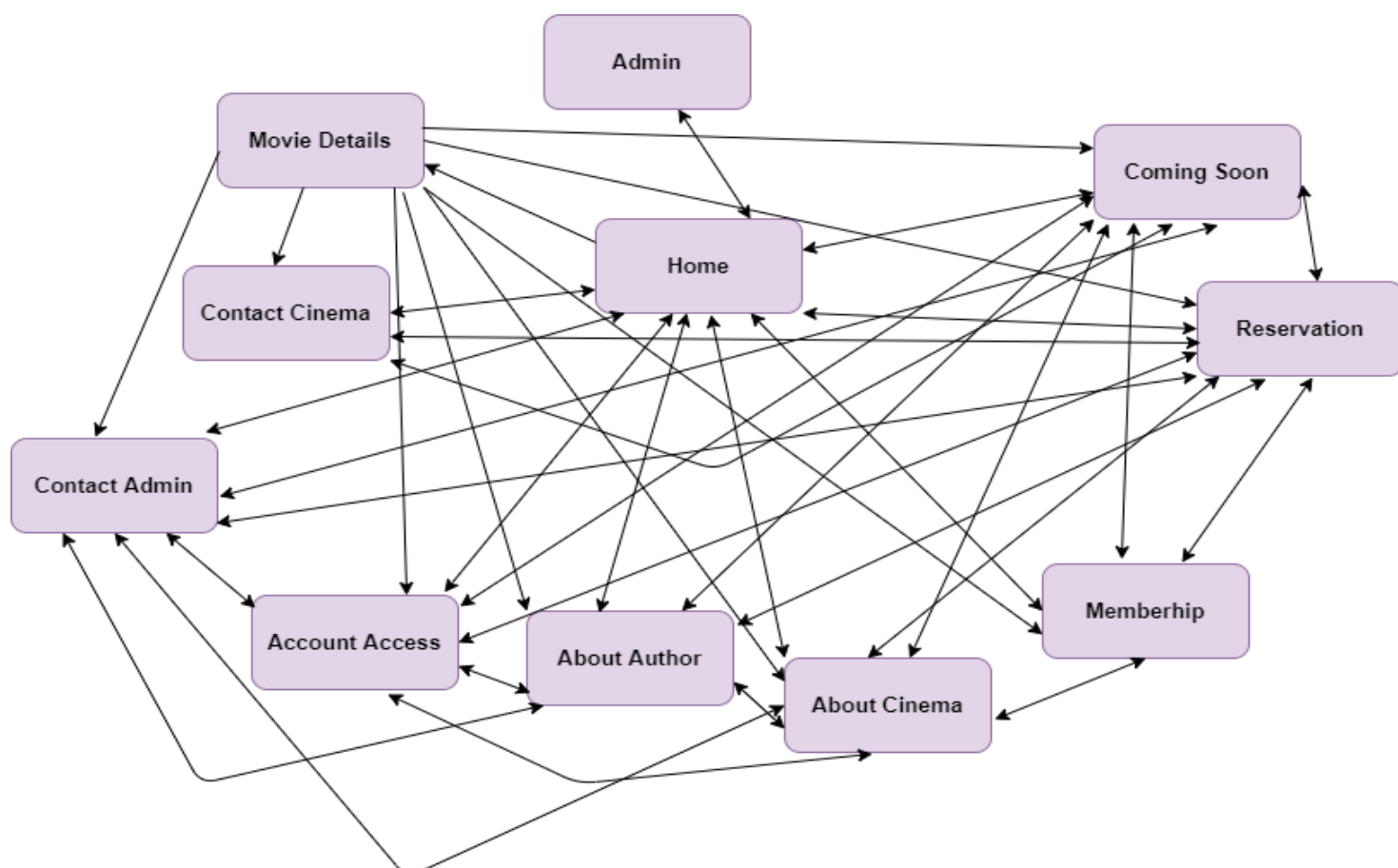
- PhpStorm
- Xampp
- phpMyAdmin

Dodaci:

- Font Awesome
- Google Fonts
- PHPMailer – Korišćen za slanje verifikacionih kodova na email, pri vraćanju pristupa nalogu, prijavi za člansku karticu bioskopa.
- Cloudinary(Laravel SDK) – obežbeđuje API za upload slika i drugih fajlova. Korisno kod hostinga kao što je Heroku koji imatu tzv. “ephemeral filesystem”.

3. ORGANIZACIJA RADA

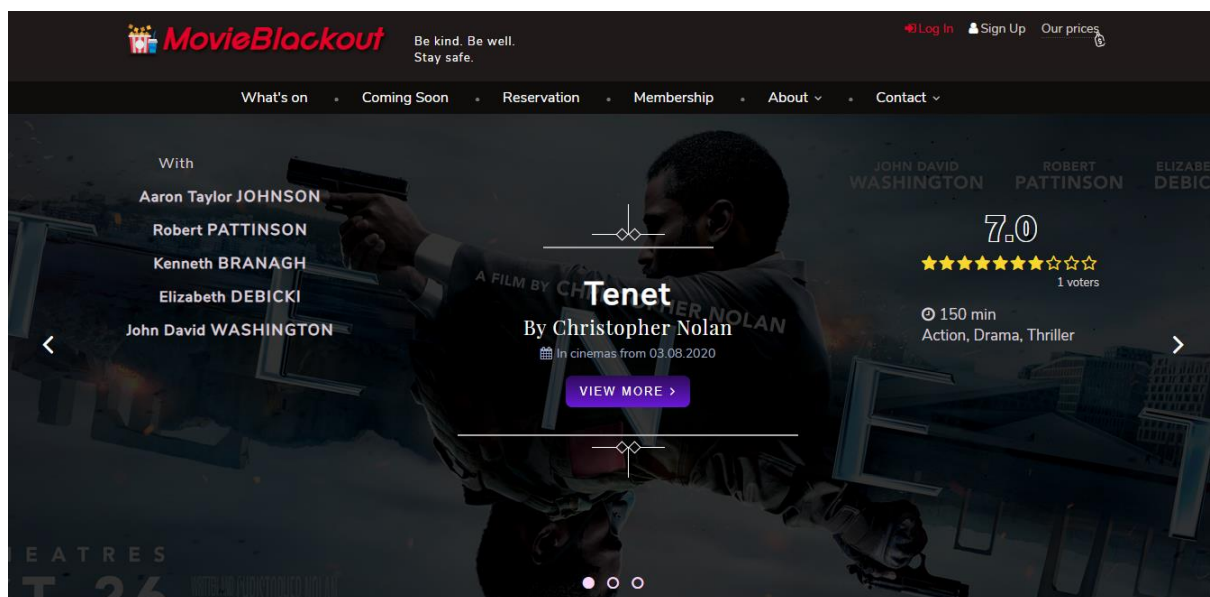
3.1 Organizaciona šema



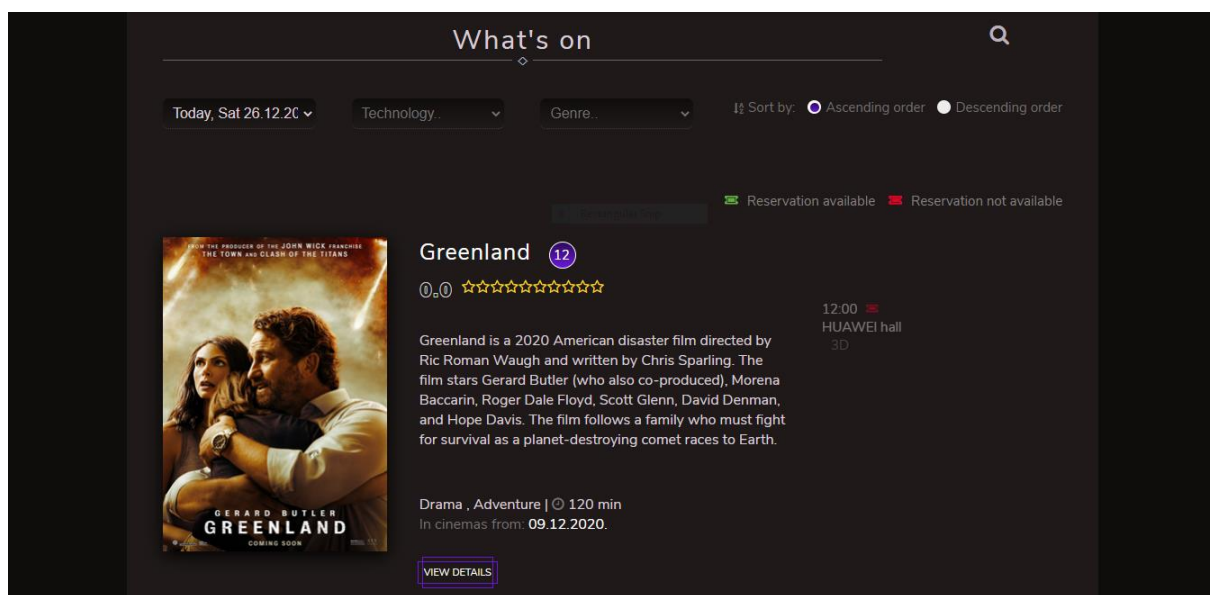
Slika 3.1.1 - Organizaciona šema

3.2 Stranice

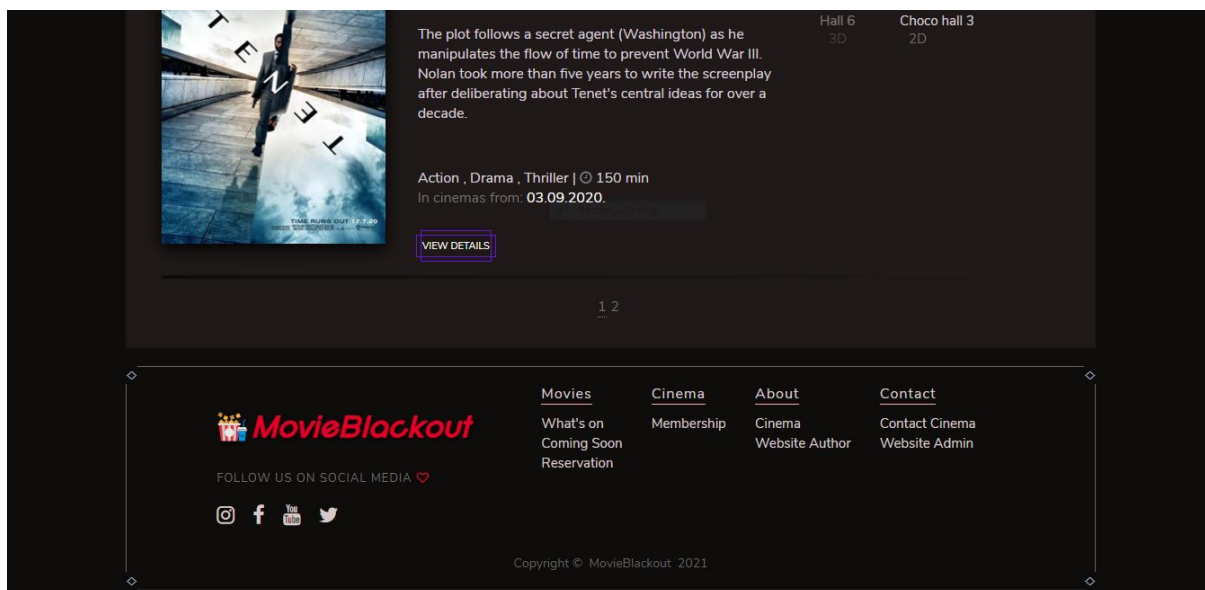
3.2.1 Home



Slika 3.2.1 - Početna stranica



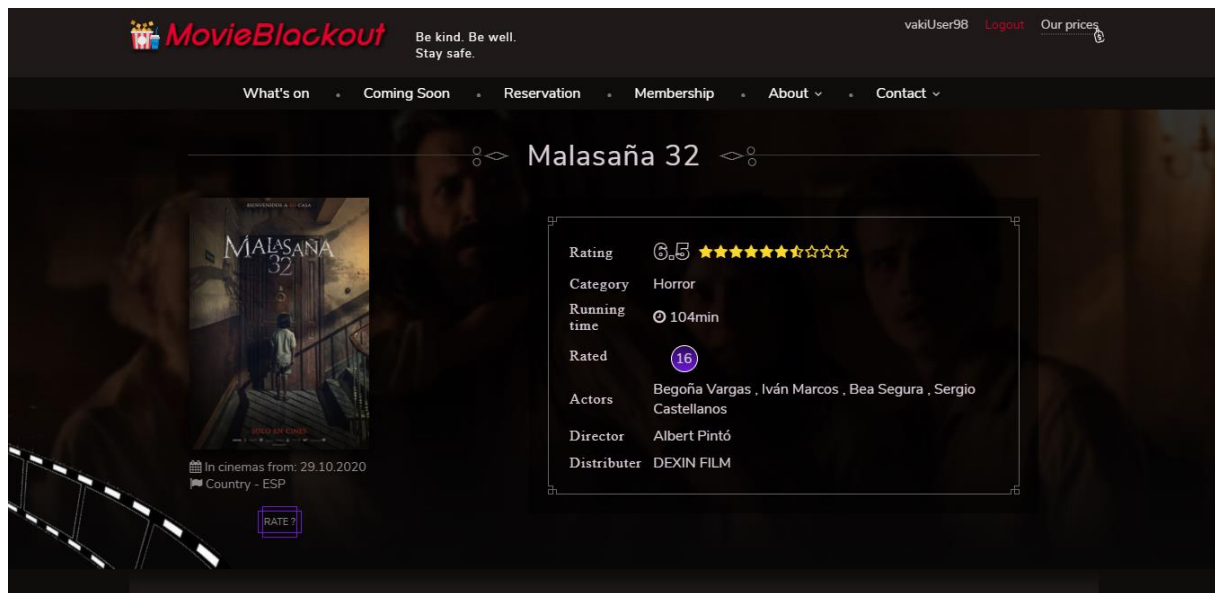
Slika 3.2.2 - Prikaz repetoara početne stranice



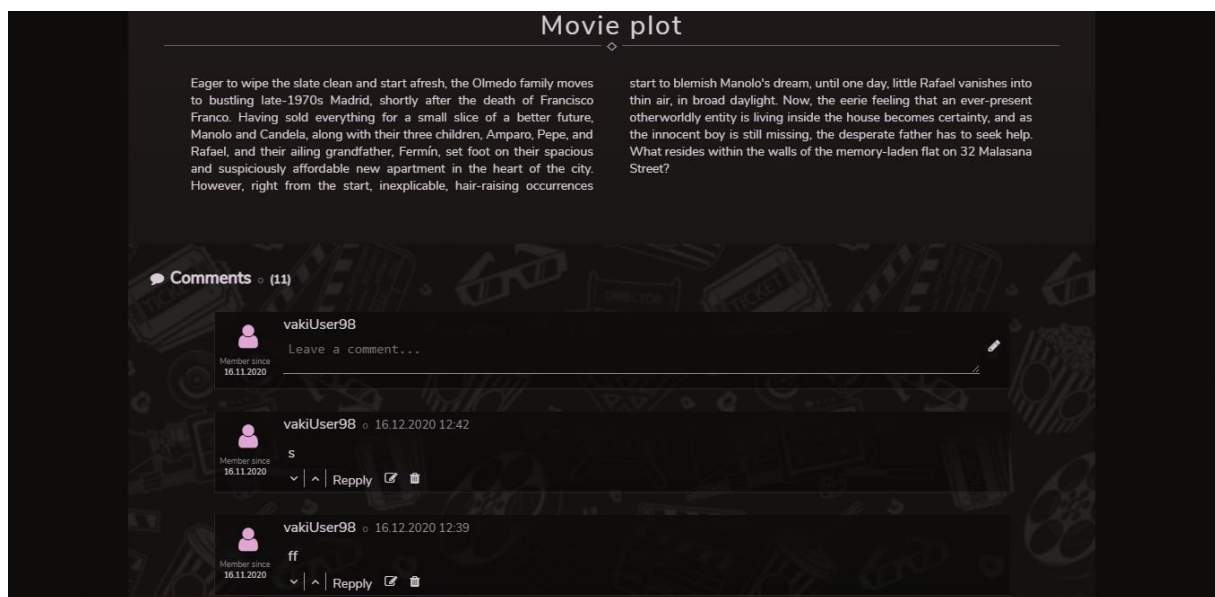
Slika 3.2.3 - Prikaz repetoara početne stranice

Korisnik ima pregled repetoara koji je trenutno u ponudi. Može vršiti filtriranje prikaza prema datumu, tehnologijama i žanrovima, pretraživati po nazivu filma i sortirati po istom, nakon čega se ispod dropdown listi pojavljuje mogućnost za brisanje svih filtera posle čega se prikaz vraća u prvobitno stanje.

3.2.2 Movie Details



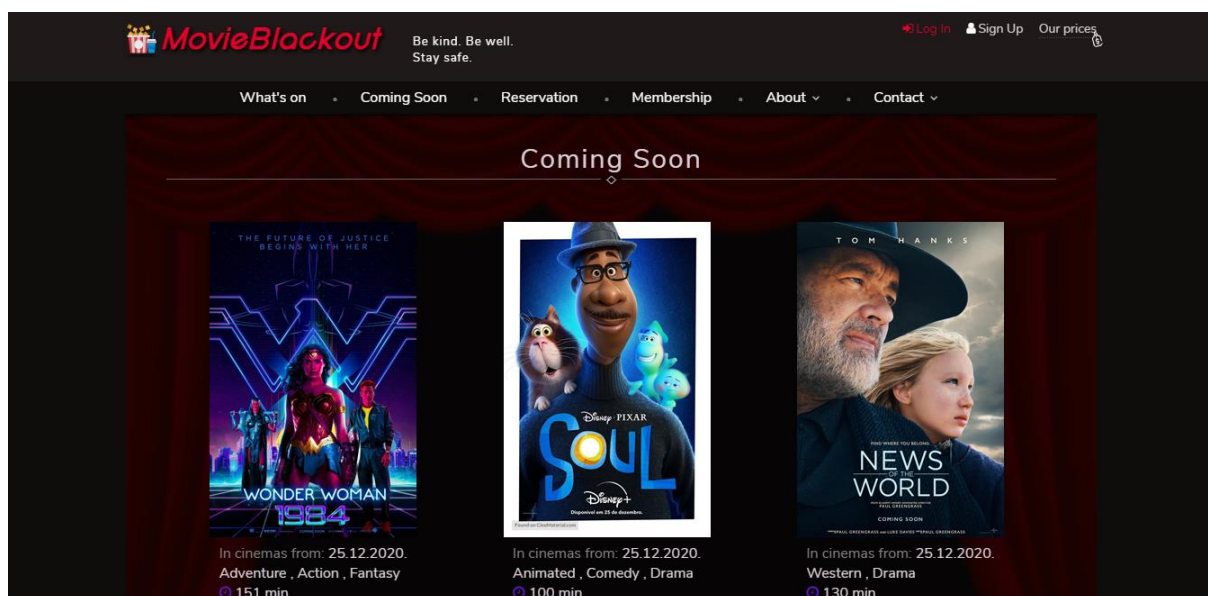
Slika 3.2.4 - Prikaz stranice sa informacijama o filmu



Slika 3.2.5 - Prikaz dela stranice sa sadržajem filma i komentarima

Stranica sa informacijama o filmu, sadrži i komentare koje mogu ostavljati samo autorizovani korisnici. Takođe može odgovoriti na nečiji komentar, editovati i brisati svoje, dati ocenu filmu koji je gledao.

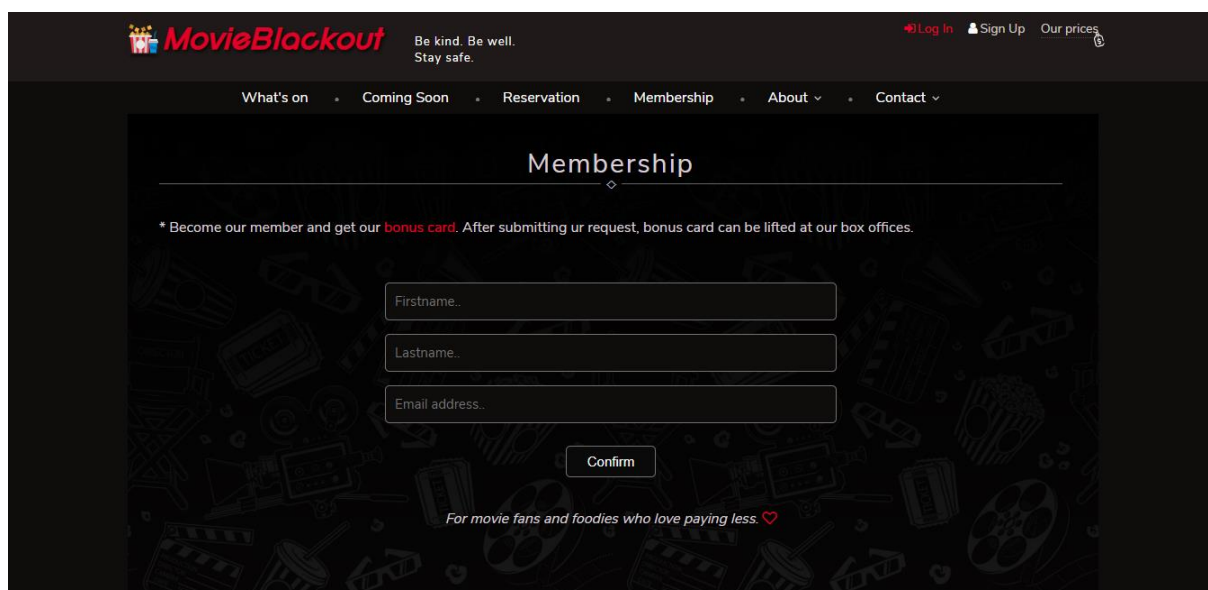
3.2.3 Cooming Soon



Slika 3.2.6 - Stranica sa filmovima koji stižu u bioskop

Stranica na kojoj korisnik može videti filmove koji će se tek davati i koji uskoro stižu u bioskop.

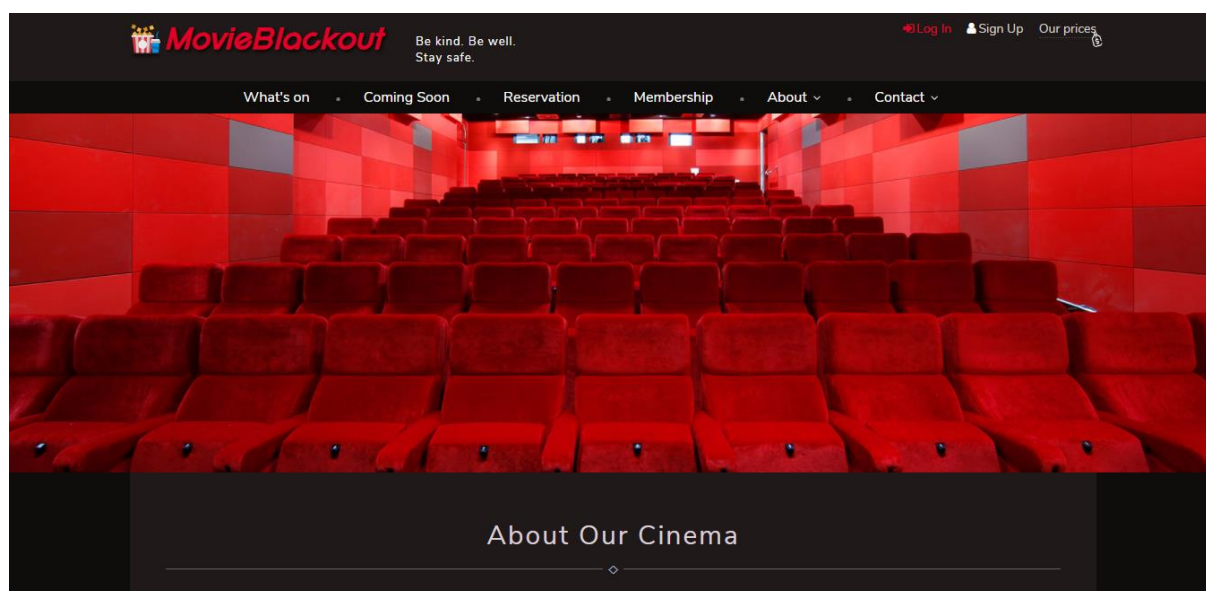
3.2.4 Membership



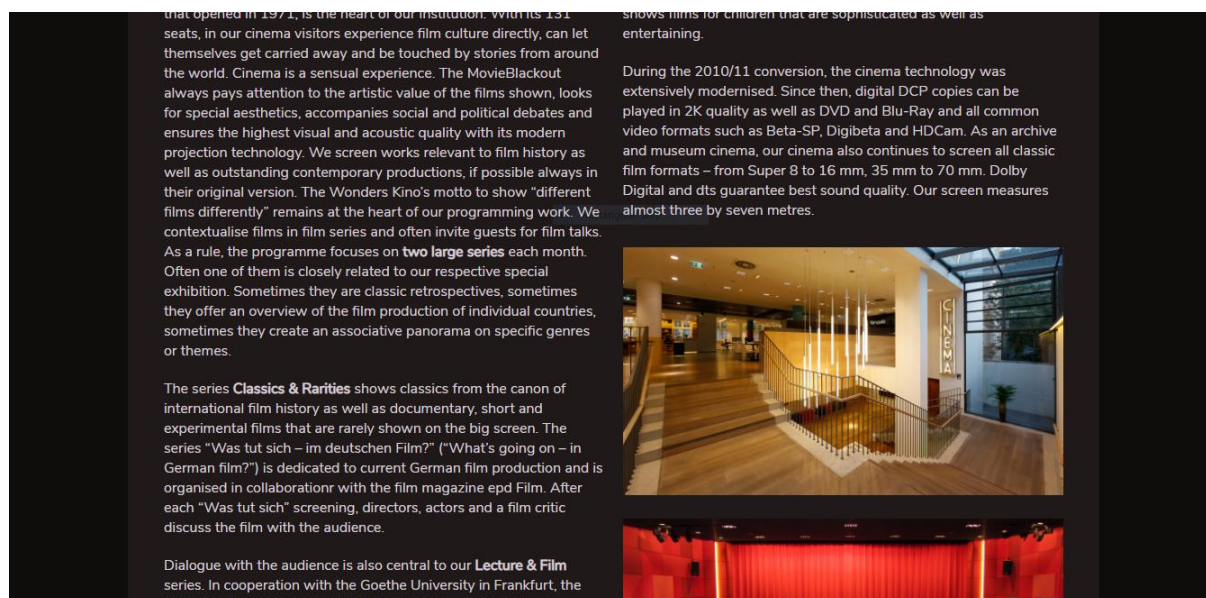
Slika 3.2.7 - Stranica za prijavu za člansku kartu

Stranica na kojoj se korisnik može prijaviti za člansku karticu. Nakon submitovanja forme na unetu email adresu se šalje verifikacioni kod koji potom treba da unese kako bi se potvrdilo da je to zaista njegova adresa.

3.2.5 About Cinema



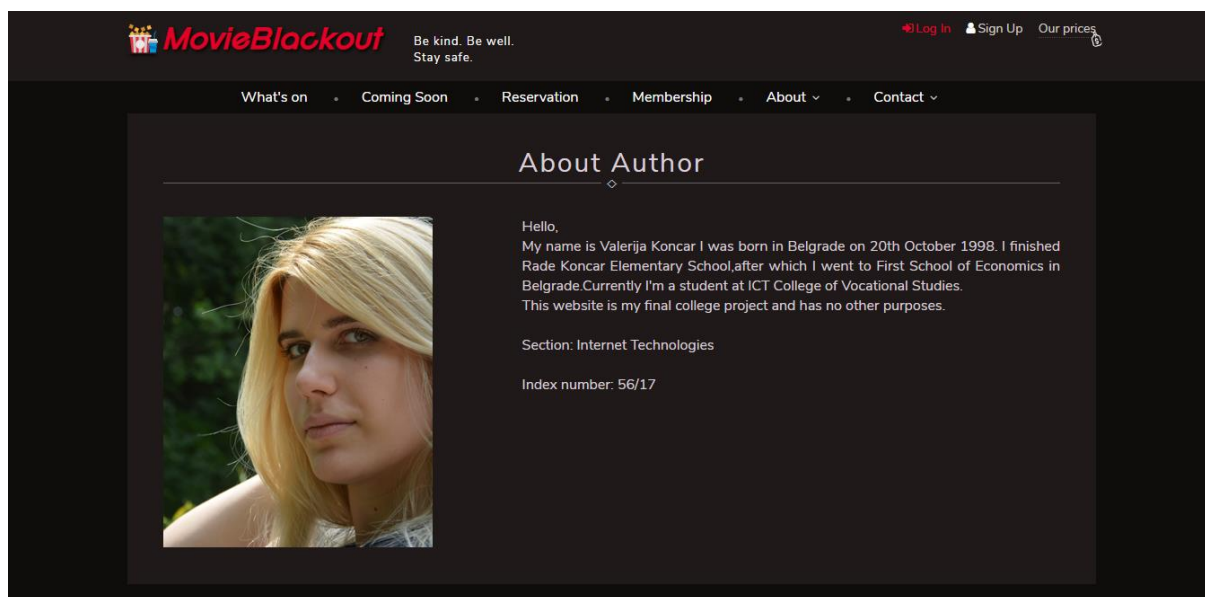
Slika 3.2.8 - Stranica o bioskopu



Slika 3.2.9 - Prikaz dela stranice sa informacijama o bioskopu

Stranica sa informacijama o bioskopu.

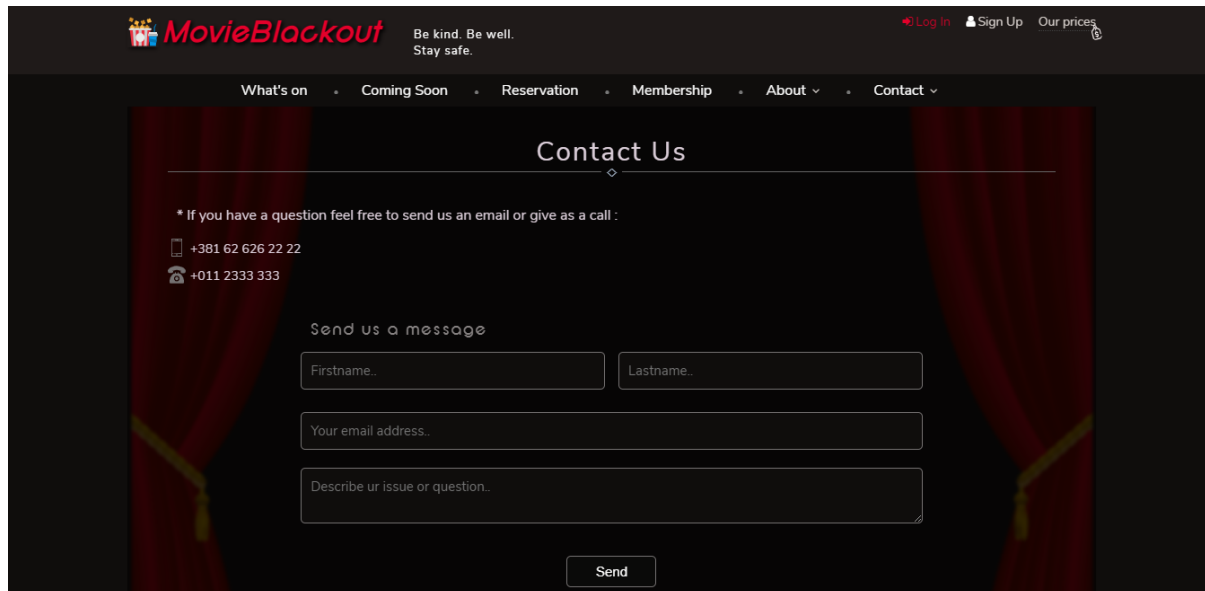
3.2.6 About Author



Slika 3.2.10 - Stranica o autoru

Stranica o autoru.

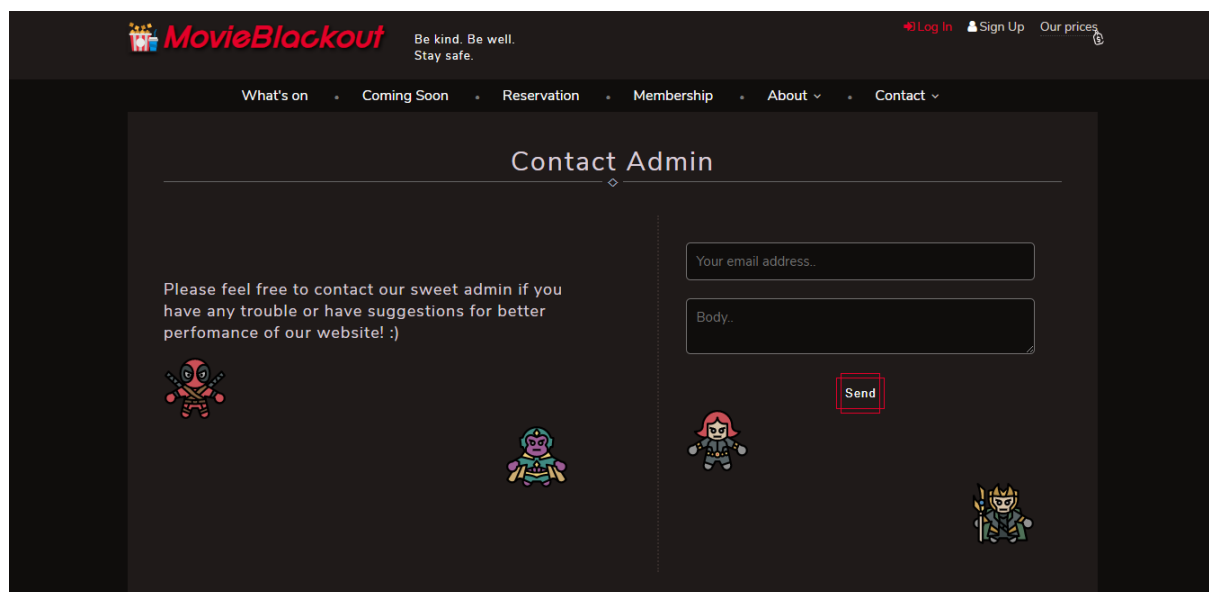
3.2.7 Contact Cinema



Slika 3.2.11 - Stranica za kontaktiranje bioskopa

Stranica sa koje korisnik može kontaktirati bioskop, ukoliko ima bilo kakvih pitanja.

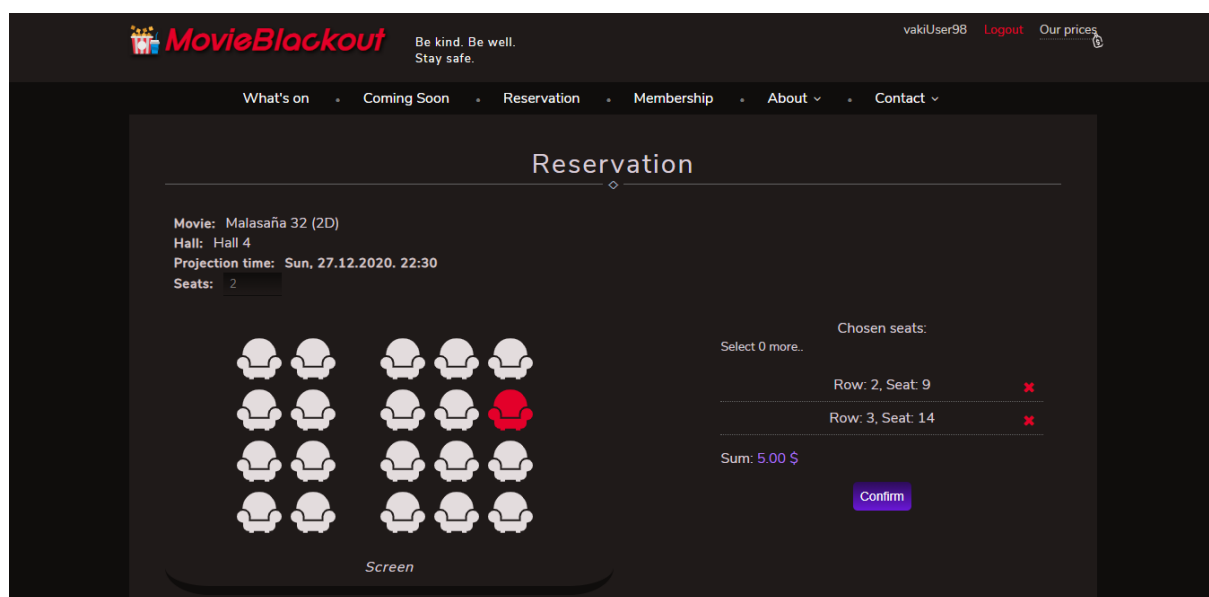
3.2.8 Contact Admin



Slika 3.2.12 - Stranica za kontaktiranje admina sajta

Stranica na kojoj korisnik može poslati poruku adminu, ukoliko ima nekih problema.

3.2.9 Reservation



Slika 3.2.13 - Stranica za rezervaciju karata

Stranica na kojoj autorizovani korisnik može rezervisati sedišta. Slobodna sedišta su bele boje, dok su zauzeta sedišta predstavljena crvenom bojom. Kako korisnik bira sedišta sa desne strane

mu se pojavljuje prikaz onih koje je izabrao i sa time se ažurira i cena. Cena karte zavisi od dana u nedelji i tehnologije u kojoj se daje izabrana projekcija. Nakon uspešne rezervacije korisnik će dobiti povratnu informaciju u vidu poruke koja potvrđuje to. Nakon toga rezervisana sedišta više nisu na raspolaganju u toj sali, za tu projekciju i biće crvene boje.

3.2.10 Regain Account Access

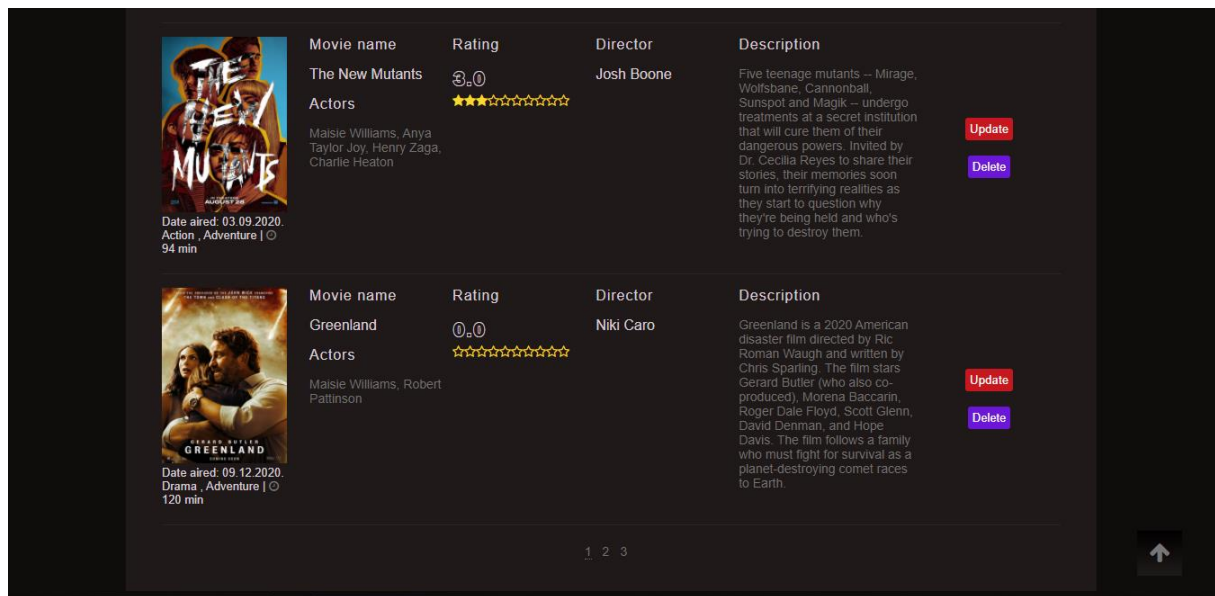
Slika 3.2.14 - Stranica za povratak pristupa nalogu

Klikom na *Forget password?* ispod forme za logovanje pojaviće se prozor u koji treba uneti email adresu sa kojom se osoba registrovala, na nju se šalje verifikacioni kod. Korisnik će biti redirektovan na stranicu za povratak pristupa nalogu gde će uneti kod i novu lozinku. Ako je sve prošlo dobro, dobiće povratnu informaciju u vidu poruke.

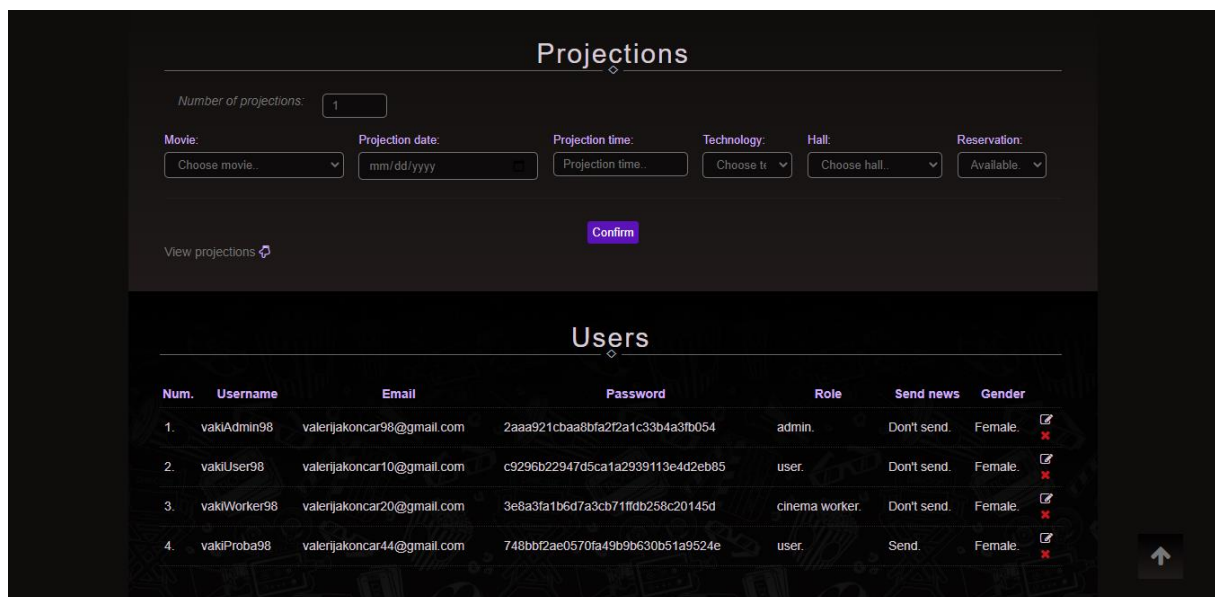
3.2.11 Admin

Movie name	Rating	Director	Description
Malasaña 32 Actors Begoña Vargas, Iván Marcos, Bea Segura, Sergio Castellanos Date aired: 29.10.2020. Horror 104 min	6.5 ★★★★★☆	Albert Pintó	Inspired by real events, the story centers around a family in the 70s who settles in the Madrid neighborhood of Malasaña where their new house will become the worst of their nightmares.
Tenet Actors	7.0 ★★★★★☆	Christopher Nolan	The plot follows a secret agent (Washington) as he manipulates the flow of time to prevent World War III. Nolan

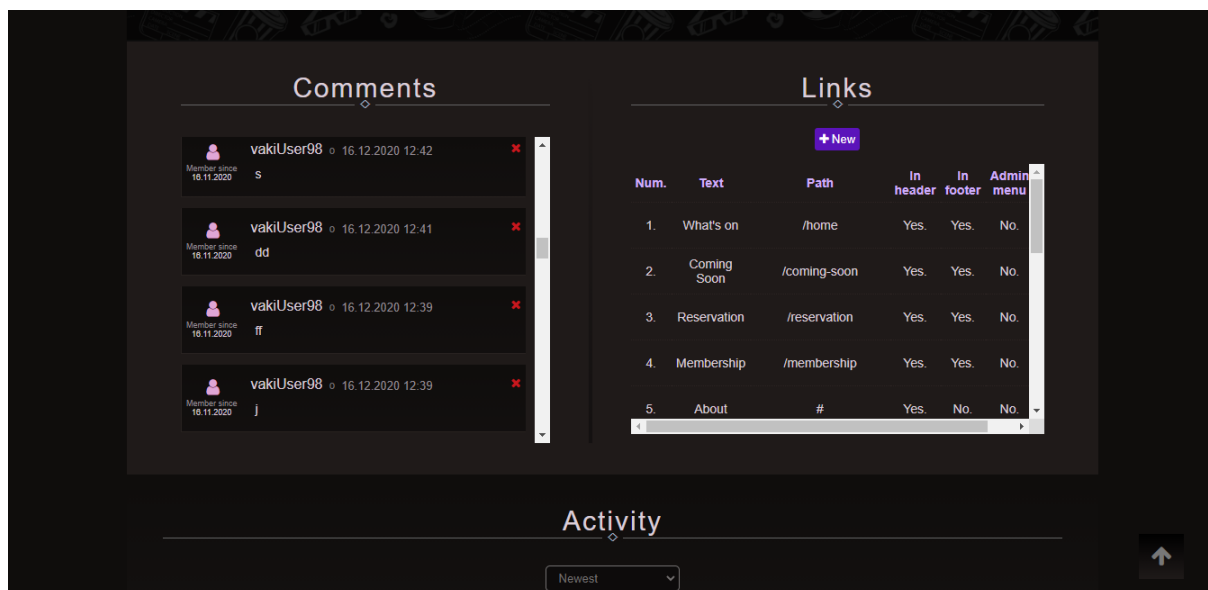
Slika 3.2.15 - Stranica za admina sajta



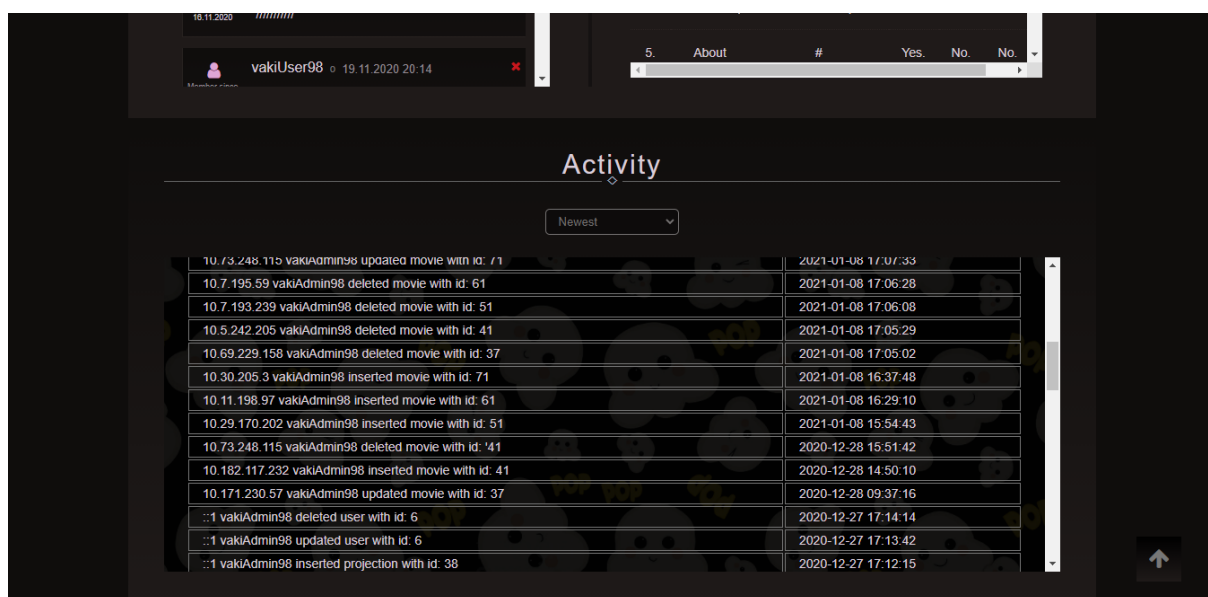
Slika 3.2.16 - Prikaz dela admin stranice za upravljanje filmovima



Slika 3.2.17 - Prikaz dela admin stranice za upravljanje projekcijama i korisnicima



Slika 3.2.18 - Prikaz dela admin str. za upravljanje komentarima i linkovima



Slika 3.2.19 - Prikaz dela admin str. sa aktivnostima admina

Ova stranica je namenjena adminu, koji može unositi nove filmove, menjati i brisati postojeće, unositi, menjati projekcije (datumi projekcija moraju biti u budućnosti), brisati ih, brisati neželjene komentare korisnika, menjati i brisati korisnike.

Svaka adminova aktivnost se beleži kako bi se imao uvid u to šta je, ko i kad radio.

4. KOD

4.1 Views

Pages

4.1.1 home.blade.php

```
@extends("template")
@section("title")
    Home
@endsection
@section("mainContent")
    <div id="slider">
        
        <div id="sliderInfoContainer">
            <div id="sliderInfoWrapper">
                <a href="#" class="arrowsSlider" id="leftArrow"><span class="fa
fa-angle-left"></span></a>
                <div id="sliderInfo">
                    <div id="sliderActors">
                    </div>
                    <div id="centerSliderInfo">
                    </div>
                    <div id="sliderGrades">
                    </div>
                </div>
                <a href="#" class="arrowsSlider" id="rightArrow"><span class="fa
fa-angle-right"></span></a>
            </div>
            <ul id="sliderCircles">
            </ul>
        </div>
    </div>
    <div id="movies">
        <div id="headlineSearchCont">
            <div id="headlineDecorationalMovies">
                <h2>What's on</h2>
                <div class="decorationalDiv">
                    <span></span>
                </div>
            </div>
            <div id="searchCont">
                <input type="text" name="search" id="extendSearch" class="extend"
placeholder="Search for movies..">
                <span class="fa fa-search searchicon" aria-hidden="true"></span>
                <span class="fa fa-close notVisible" aria-hidden="true"></span>
            </div>
            <div class="cleaner"></div>
        </div>
        <div id="filterSortMovies">
            <?php date_default_timezone_set('Europe/Belgrade');?>
            <?php
            $todayDate = date("d.m.Y.", strtotime("2020-12-26"));
            $user = 0;
            if(session()->has("user")){
                $user = 1;
            }
            ?>
```



```

        @endforeach
    </div>
    <div id="paginationDiv">
        <ul id="pagination">
            <?php
            for($i = 0; $i < $pagination; $i++):
            ?>
            <li>
                <a href="#" class="paginationLinks <?php if($i == 0){ echo
"activePagination";}>" data-limit="<?=$i ?>"><?=$i+1 ?></a>
            </li>
            <?php endfor; ?>
        </ul>
    </div>
</div>

@endsection

```

4.1.2 movie_details.blade.php

```

@extends("template")
@section("title")
    {{$movie->movieName}}
@endsection
@section("mainContent")
    <div id="backgroundPic" style="background-image: url('{{asset($movie-
>backgroundPicName->path.$movie->backgroundPicName->name)}}');">
        <div id="transparentDivOnBg">
            <div id="abtMovie">
                <div class="fancyTitle">
                    <div class="fancyTitlePre">
                        <div class="line"></div>
                        <div class="end"></div>
                    </div>
                    <div class="fancyTitleInner">
                        <h2>{{$movie->movieName}}</h2>
                    </div>
                    <div class="fancyTitlePost">
                        <div class="end"></div>
                        <div class="line"></div>
                    </div>
                </div>
                <div id="movieInformation">
                    <div id="leftMovieInformation">
                        alt }}" width="200px" height="300px"/>
                        <?php
                        $toDate = date_create($movie->in_cinemas_from);
                        $formattedDate = date_format($toDate, 'd.m.Y');
                        ?>
                        <p><span class="fa fa-calendar"></span> In cinemas from:
                        {{$formattedDate}}</p>
                        <p><span class="fa fa-flag"></span> Country - {{$movie-
>country}}</p>
                        <?php if(session()->has('user')){ ?>
                        {{--
                        && session("user")->role_id === 1--}}

```

```

{{--
    <a href="#" class="vote">Rate?</a>--}}
    <a href="#" class="detailsLink vote" data-movid="{{ $movie-
>idMovie }}" id="abtMovieBook">Rate ?</a>
    <div class="voteDiv">
        <div class="giveVote">
            <div class="faCloseRateCloud"><span class="fa fa-
close"></span></div>

            <p>{{ $movie->movieName }}</p>
            <ul class="giveVoteList">
                @for($i = 0; $i < 10; $i++)
                    <li><span class="fa fa-star-o" data-
index="{{ $i }}"></span></li>
                @endfor
            </ul>
            <span class="userChosenGrade">0.0</span>
            <input type="button" value="Rate"
class="confirmVote" data-movid="{{ $movie->idMovie }}" />
        </div>
    </div>
    <?php } ?>
</div>
<div id="rightMovieInformation">
    <div id="rightMovieInformationTransparent">
        <div class="fancy-box">
            <span class="fancy-box__tl"></span>
            <span class="fancy-box__tr"></span>
            <span class="fancy-box__bl"></span>
            <span class="fancy-box__br"></span>
            <span class="fancy-box__sides"></span>
            <div class="fancy-box__inner">
                <table id="tableInfo">
                    <tr>
                        <td>Rating</td>
                        <td class="ratingTd">
                            <p class="numberMovieGrade
numberMovieGradeMovDet">{{ $movie->grade }}</p>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            @component("partials.movie_stars", ["movie" => $movie])
                                @endcomponent
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Category
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <?php $i=0; ?>
                            @foreach($movie->genres as
$genre)
                                <?php $i++; ?>
                                {{ $genre->genre }}
                                @if(count($movie->genres
!= $i)
                                    {{ " , " . " " }}
                                @endif
                            @endforeach
                        </td>
                    </tr>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

                                <td>Running time</td>
                                <td><span class="fa fa-clock-
o"></span> {{ $movie->running_time }}min</td>
                                </tr>
                                <tr>
                                <td>Rated</td>
                                <td><p class="ageLimit">{{ $movie-
>age_limit }}</p></td>
                                </tr>
                                <tr>
                                <td>Actors</td>
                                <td id="actorsTd">
                                    <?php $i = 0;?>
                                    @foreach($movie->actors as $a)
                                        <?php $i++; ?>
                                        {{ $a->name }}
                                        @if(count($movie-
>actors) != $i)
                                            {{", "}}
                                        @endif
                                    @endforeach
                                </td>
                                </tr>
                                <tr>
                                <td>Director</td>
                                <td>{{ $movie->director }}</td>
                                </tr>
                                <tr>
                                <td>Distributer</td>
                                <td>{{ $movie->distributer }}</td>
                                </tr>
                                </table>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <div id="tapeHolder">
                <div class="tape">
                    <div class="tapeHold"></div>
                </div>
            </div>
        </div>
        <div id="centeredWrapper">
            <div id="plot">
                <h2>Movie plot</h2>
                <div class="decorationalDiv">
                    <span></span>
                </div>
                <div id="plotText">
                    {{ $movie->description_detailed }}
                </div>
            </div>
        </div>
        <div id="comments">
            <div id="commentsTransparent">

```

```

        <h3><span class="fa fa-comment"></span>Comments<span
id="numOfComments">(<span id="commentsNumber"><?php echo($movie-
>commentsCount);?></span></span></h3>
        <div id="peopleComments">
            @if(session()->has('user'))
                <form id="postCommentForm" action="#" method="POST">
                    @csrf
                    <div id="leaveComm">
                        <div class="userFaHolder">
                            @if(session('user')->role_id == 2)
                                <p class="adminComment">Admin</p>
                            @endif
                            <span class="fa fa-user
                                @if(session('user')->gender_id == 1)
                                    femaleUser
                                @else
                                    maleUser
                                @endif
                                "></span>
                            <p class="memberSince">Member since</p>
                                <?php
                                    $toDate = date_create(session('user')-
>registered_at);
                                    $formattedDate = date_format($toDate,
'd.m.Y');
                                ?>
                                <p class="memberDate">{{{$formattedDate}}}</p>
                            </div>
                            <div class="commentRight">
                                <div class="topPartComment">
                                    <span
class="usernameComment">{{session('user')->username}}</span>
                                </div>
                                <textarea id="myComment" placeholder="Leave a
comment..."></textarea>
                                </div>
                                <span class="fa fa-pencil" data-movid="{{{$movie-
>idMovie}}}"></span>
                            </div>
                        </form>
                    @endif
                    @foreach($movie->comments as $c)
                        @component("partials.comment",["c" => $c])
                        @endcomponent
                    @endforeach
                </div>
                <a href="#" id="viewMoreCom" data-movid="{{{$movie->idMovie}}}">View
more ...</a>
            </div>
        </div>
    @endsection

```

4.1.3 coming_soon.blade.php

```

@extends("template")
@section("title")
    Cooming Soon
@endsection
@section("mainContent")

```



```

<div id="centeredComingSoon">
    <div id="curtain">

    </div>
    <div id="csHolderOpacity">
        <h2>Coming Soon</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <div id="csMovieHolder">
            @foreach($coming_soon as $cs)
                <div class="cs_movie">
                    alt }}" class="imgCS" width="230" height="350"/>
                    <?php
                        $toDate = date_create($cs->in_cinemas_from);
                        $formattedDate = date_format($toDate, 'd.m.Y. ');
                    <?>
                    <div class="csInfoHolder">
                        <p class="inCinemasCS"><span>In cinemas from:</span>
                        {{ $formattedDate }}</p>
                        <p class="genresCS">
                            <?php $i=0; ?>
                            @foreach($cs->genres as $genre)
                                <?php $i++; ?>
                                {{ $genre->genre }}
                                @if(count($cs->genres) != $i)
                                    {{ " , " . " " }}
                                @endif
                            @endforeach
                        </p>
                        <p><span class="fa fa-clock-o"></span> {{ $cs->running_time }} min</p>
                    </div>
                </div>
            @endforeach
        <div class="cleaner"></div>
    </div>
</div>
@endsection

```

4.1.4 Membership.blade.php

```

@extends("template")
@section("title")
    Membership
@endsection
@section("mainContent")
    <div id="centeredMembershipBg">
        <div id="centeredMembership">
            <h2>Membership</h2>
            <div class="decorationalDiv">
                <span></span>
            </div>
            <p class="membershipNotice" id="membershipNotice1">* Become our member
and get our <span>bonus card</span>. After submitting ur request, bonus card can
be lifted at our box offices. </p>
            <div id="membformDiv">

```

```

        @if(session()->has('successSentEmailMembership'))
            <p class="success"
class="noticeSuccErrMembership">{{session('successSentEmailMembership')}}</p>
        @endif
        @if(session()->has('errorSentEmailMembership'))
            <p class="error"
class="noticeSuccErrMembership">{{session('errorSentEmailMembership')}}</p>
        @endif
        @if(session()->has('successSentEmailMembership'))
            <?php
                $userInfo = null;
                if(session()->has('userMembershipInfo')){
                    $userInfo = session('userMembershipInfo');
                }
            ?>
            <form action="{{url('/becomeMemberRequest')}}"
id="membershipRequestForm" method="post">
                @csrf
                <div>
                    <input type="text" id="memReqName"
name="memReqName" placeholder="Firstname.."
value="{{ $userInfo? $userInfo['firstname']: '' }}"/>
                    <p class="error" id="memReqNameErr"></p>
                </div>
                <div>
                    <input type="text" id="memReqLastname"
name="memReqLastname" placeholder="Lastname.."
value="{{ $userInfo? $userInfo['lastname']: '' }}"/>
                    <p class="error" id="memReqLastnameErr"></p>
                </div>
                <div>
                    <input disabled type="text" id="memReqEmail"
name="memReqEmail" placeholder="Email address.."
value="{{ $userInfo? $userInfo['email']: '' }}"/>
                    <p class="error" id="memReqEmailErr"></p>
                </div>
                <div>
                    <input type="text" id="memReqCode"
name="memReqCode" placeholder="Received verification code.."/>
                    <p class="error" id="memReqCodeErr"></p>
                </div>
                <input type="hidden" id="memReqUserId"
name="memReqUserId" value="{{ $userInfo? $userInfo['userId']: '' }}"/>
                <div><input type="submit" id="sumbitReqMembership"
name="sumbitMembership" value="Confirm"/></div>
            </form>

        @else

            <form action="{{url('/becomeMember')}}"
id="membershipForm" method="post">
                @csrf
                <?php
                    $user = null;
                    if(session()->has('user')){
                        $user = session('user');
                    }
                ?>
                <div><input type="text" id="memName" name="memName"

```

```

placeholder="Firstname.."/></div>
        <div><input type="text" id="memLastname"
name="memLastname" placeholder="Lastname.."/></div>
        <div><input type="text" id="memEmail" name="memEmail"
placeholder="Email address.." value="{{ $user ? $user->email : '' }}"></div>
        <input type="hidden" id="membershipHidden"
name="membershipHidden" value="{{ $user ? $user->id : '' }}">
        <div><input type="submit" id="sumbitMembership"
name="sumbitMembership" value="Confirm"/></div>
        @isset($errors)
            <p class="error">
                @foreach($errors->all() as $error)
                    {{ $error }}<br/>
                @endforeach
            </p>
        @endisset
        @if(session()->has('error'))
            <p class="error">
                {{ session('error') }}
            </p>
        @endif
        @if(session()->has('success'))
            <p class="success">
                {{ session('success') }}
            </p>
        @endif
    </form>
    @endif
</div>
    <p class="membershipNotice" id="membershipNotice2">For movie fans and
foodies who love paying less. <span class="fa fa-heart-o"></span></p>
</div>
</div>
@endsection

```

4.1.5 about_cinema.blade.php

```

@extends("template")
@section("title")
    About Cinema
@endsection
@section("mainContent")
    <div id="aboutCinemaSlider">

    </div>
    <div id="centeredAboutCinema">
        <h2>About Our Cinema</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <div id="abtCinemaInfo">
            <?php echo $text; ?>
            
            
        </div>
    </div>
    @endsection

```

4.1.6 about_author.blade.php

```
@extends("template")
@section("title")
About Author
@endsection
@section("mainContent")
<div id="centeredAboutAuthor">
    <h2>About Author</h2>
    <div class="decorationalDiv">
        <span></span>
    </div>
    <div id="author">
        <div id="picAuthor">
            
        </div>
        <div id="authorAbout">
            <?= $author ?>
        </div>
    </div>
</div>
</div>
@endsection
```

4.1.7 contact_cinema.blade.php

```
@extends("template")
@section("title")
    Contact Us
@endsection
@section("mainContent")
    <div id="centeredContactCinema">
        <div id="curtain">

        </div>
        <div id="contactCinemaHolderOpacity">
            <h2>Contact Us</h2>
            <div class="decorationalDiv">
                <span></span>
            </div>
            <div id="contCinNoticeDiv">
                <p id="contCinNotice">* If you have a question feel free to send
us an email or give as a call :</p>
                <div class="phoneHolder"><p>+381 62 626 22
22</p></div>
                <div class="phoneHolder"><p>+011 2333 333</p></div>
                </div>
                <div id="contactCinemaHolder">
                    <p id="contCinSendMail">Send us a message</p>
                    <form id="contactCinema" action="{{url('/contact-cinema')}}"
method="post">
                        @csrf
                    </form>
                </div>
            </div>
        </div>
    </div>
@endsection
```

```

        <div id="nameHolderContCinema">
            <input type="text" id="firstnameContCin"
name="firstnameContCin" placeholder="Firstname.."/>
            <input type="text" id="lastnameContCin"
name="lastnameContCin" placeholder="Lastname.."/>
        </div>
        <p class="error" id="firstnameContCinErr"></p>
        <p class="error" id="lastnameContCinErr"></p>
        <input type="text" id="emailAddressContCin"
name="emailAddressContCin" placeholder="Your email address.."
value="@if(session()->has('user')){{session('user')->email}}@endif" />
        <p class="error" id="emailAddressContCinErr"></p>
        <textarea id="question" name="question" placeholder="Describe
ur issue or question.."></textarea>
        <p class="error" id="questionErr"></p>
        @if(session()->has("success"))
            <p class="success">{{session("success")}}</p>
        @endif
        @if(session()->has("error"))
            <p class="error">{{session("error")}}</p>
        @endif
        <div id="sendContCinDiv">
            <input type="submit" value="Send" id="sendContCin"/>
        </div>
    </form>
</div>
</div>
</div>
@endsection

```

4.1.8 contact_admin.blade.php

```

@extends("template")
@section("title")
Contact Admin
@endsection
@section("mainContent")
<div id="centeredContactAdmin">
    <h2>Contact Admin</h2>
    <div class="decorationalDiv">
        <span></span>
    </div>
    <div id="contact">
        <div id="contactAdminDiv">
            <div id="adm">
                Please feel free to contact our sweet admin if you have any
trouble or have suggestions for better
performance of our website! :)
                <div id="deadpool"></div>
                
                <div class="cleaner"></div>
            </div>
            <form id="contactAdmin" action="{{url('/contact-admin')}}"
method="post">
                @csrf
                <input type="text" id="emailAddress" name="emailAddress"
placeholder="Your email address.." value="@if(session()-

```

```

>has('user')){{session('user')->email}}@endif" />
    <p class="error" id="emailAddressErr"></p>
    <textarea id="body" name="body" placeholder="Body.."></textarea>
    <p class="error" id="bodyErr"></p>
    @if(session()->has("success"))
        <p class="success">{{session("success")}}</p>
    @endif
    @if(session()->has("error"))
        <p class="error">{{session("error")}}</p>
    @endif
    <div id="sendAdminDiv">
        <input type="submit" value="Send" id="sendAdmin"/>
    </div>
    
    <p class="success"></p>
    <p class="error"></p>
    
    <div class="cleaner"></div>
</form>
</div>
</div>
</div>
@endsection

```

4.1.9 reservation.blade.php

```

@extends("template")
@section("title")
    Reservation
@endsection
@section("mainContent")
    <div id="resWrapper">
        <h2>Reservation</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <ul id="projectionInfo">
            <li><span>Movie:</span> {{ $projectionInfo->movieName }}
            ({{ $projectionInfo->technName }})</li>
            <li><span>Hall:</span> {{ $projectionInfo->hallName }}</li>
            <li><span>Projection time:</span> <span><?= date("D, d.m.Y. H:i",
strtotime($projectionInfo->starts_at)); ?></span></li>
            <li><span>Seats:</span> <span><input type="number" id="inputSeatsNum"
value="{{ $seats }}" min="1" max="10"/></span></li>
            {{ --
            <span>{{ $seats }}</span> -- }}
        </ul>
        <input type="hidden" value="{{ $projectionInfo->idProjection }}"
id="projectionId" name="projectionId"/>
        <input type="hidden" value="{{ $pricePerTicket }}" id="pricePerTicket"
name="pricePerTicket"/>
        <div id="hallSeatsHolder">
            <div id="hallSeats">
                @foreach($hallRows as $hr)
                    <div class="hallRow" data-id="{{ $hr->id }}" data-row="{{ $hr->
row }}">
                        @foreach($hr->seats as $s)
                            @if($s->is_available)

```



```

        <form id="accountAccessForm" action="{{ url('/account-access')
}}}" method="POST">
            @csrf
            <p id="helloAccountAccess">Hello {{session("username")-
>username}}!</p>
            <input type="text" id="accountAccessEmail"
name="accountAccessEmail" placeholder="Your email address.." value="{{session()-
>has('email')?session('email'):''}}" disabled/>
            <p class="error" id="accountAccessEmailErr"></p>
            <input type="text" id="accountAccessCode"
name="accountAccessCode" placeholder="Recieved code.."/>
            <p class="error" id="accountAccessCodeErr"></p>
            <input type="password" id="accountAccessPass"
name="accountAccessPass" placeholder="New password.."/>
            <p class="error" id="accountAccessPassErr"></p>
            <p class="error">
                @if(session()->has('error'))
                    {{session('error')}}
                @endif
            </p>
            <p class="success">
                @isset($successfullyChangedPass)
                    {{$successfullyChangedPass}}
                @endisset
            </p>
            <input type="submit" id="accountAccessSubmit"
name="accountAccessSubmit" value="Confirm"/>
        </form>
    </div>
</div>
</div>
</div>
@endsection

```

Admin/pages

4.1.11 admin.blade.php

```

@extends("admin.admin_template")
@section("title")
    Admin
@endsection
@section("mainContent")
    <div id="centeredAdmin">
        <div id="adminMovies">
            <h2>Movies</h2>
            <div class="decorationalDiv">
                <span></span>
            </div>
            <button id="insertMovie"><span class="fa fa-plus"></span>New</button>
            <div id="searchContAdm">
                <input type="text" name="search" id="extendSearchAdm"
class="extended" placeholder="Search for movies..">
                <span class="fa fa-search searchicon" aria-hidden="true"></span>
                <span class="fa fa-close notVisible" aria-hidden="true"></span>
            </div>
            <div class="cleaner"></div>
            <div id="adminMoviesHolder">

```



```

        @foreach($movies as $m)
            @component("admin.partials.movie",["m" => $m])
                @endcomponent
            @endforeach
        </div>
        <div id="paginationDiv" class="paginationAdminDiv">
            <ul id="pagination">
                <?php
                for($i = 0; $i < $pages; $i++):
                ?>
                <li>
                    <a href="#" class="<?php if($i == 0){ echo
"activePagination";}> paginationAdmMov" data-limit="<?= $i ?>"><?= $i+1 ?></a>
                </li>
                <?php endfor; ?>
            </ul>
        </div>
    </div>
    <div id="adminProjections">
        <h2>Projections</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <div id="projectionsNumDiv">
            <p id="projNumP">Number of projections:</p>
            <input type="number" id="numOfProjections" name="numOfProjections"
class="inputAdminClass" min="1" max="5" value="1"/>
        </div>
        <div id="projectionsToInsertDiv">
            <form method="POST" action="{{url('insertProjections')}}">
id="projectionsForm" name="projectionsForm">
                @csrf
                <div class="adminProjection">
                    <div class="projectionRow">
                        <div class="projectionMovieLabel projectionLabel">
                            <label class="projectionLab">Movie:</label>
                            <select name="projectionMovie[]"
class="projectionMovie inputAdminClass">
                                <option value="0">Choose movie..</option>
                                @foreach($allMovies as $m)
                                    <option value="{{{$m->id}}}">{{{$m-
>name}}}</option>
                                @endforeach
                            </select>
                        </div>
                        <div class="projectionDateLabel projectionLabel">
                            <label class="projectionLab">Projection
date:</label>
                            <input type="date" name="projectionDate[]"
class="projectionDate inputAdminClass"/>
                        </div>
                        <div class="projectionTimeLabel projectionLabel">
                            <label class="projectionLab">Projection
time:</label>
                            <input type="text" name="projectionTime[]"
class="projectionTime inputAdminClass" placeholder="Projection time.."/>
                        </div>
                        <div class="projectionTechnologyLabel
projectionLabel">

```

```

        <label class="projectionLab">Technology:</label>
        <select name="projectionTechnology[]"
class="projectionTechnology inputAdminClass">
            <option value="0">Choose technology.</option>
        </select>
    </div>
    <div class="projectionHallLabel projectionLabel">
        <label class="projectionLab">Hall:</label>
        <select name="projectionHall[]"
class="projectionHall inputAdminClass">
            <option value="0">Choose hall.</option>
            @foreach($halls as $h)
                <option value="{{ $h->id }}">{{ $h->
>name}}</option>
            @endforeach
        </select>
    </div>
    <div class="projectionResAvailableLabel
projectionLabel">
        <label class="projectionLab">Reservation:</label>
        <select name="projectionResAvailable[]"
class="projectionResAvailable inputAdminClass">
            <option value="0">Not available.</option>
            <option value="1"
selected="true">Available.</option>
        </select>
    </div>
    <div class="cleaner"></div>
</div>
<p class="error"></p>
</div>
@isset($errors)
    <p class="error">
        @foreach($errors->all() as $err)
            {{ $err }}
        @endforeach
    </p>
@endisset
@if(session()->has('errorProjectionInsert'))
    <p class="error">
        {{session("errorProjectionInsert")}}
    </p>
@endif
@if(session()->has('successProjectionInsert'))
    <p class="success">
        {{session("successProjectionInsert")}}
    </p>
@endif
    <input type="submit" value="Confirm" id="projectionsInsert"
name="projectionsInsert"/>
</form>
    <a href="#" id="seeProjections">
        View projections <span class="fa fa-hand-o-down"></span>
    </a>
</div>
<div id="viewProjections">
    <input type="search" placeholder="Movie name.."
id="searchProjections" name="searchProjections"/>
    <div id="admProj">

```

```

        <div id="projectionLabels">
            <label>Movie</label>
            <label>Projection time</label>
            <label>Technology</label>
            <label>Hall</label>
            <label>Reservation</label>
            <label class="lastPr"></label>
        </div>
        <div id="projectionViews">
            @foreach($projections as $p)
                <?php
                    $toDate = date_create($p->starts_at);
                    $formattedDate = date_format($toDate, 'd.m.Y. H:i');
                    ?>
                <div class="projectionViewRow">
                    <div class="prMovie prInfo">{{ $p->name }}</div>
                    <div class="prDate
prInfo">{{ $formattedDate }}</div>
                    <div class="prTechn prInfo">{{ $p-
>technName }}</div>
                    <div class="prHall prInfo">{{ $p->hallName }}</div>
                    <div class="prRes prInfo">
                        @if($p->reservation_available)
                            Available.
                        @else
                            Not available.
                        @endif
                    </div>
                    <div class="prUPdDel prInfo lastPr">
                        <span class="fa fa-edit" data-id="{{ $p-
>id }}"></span>
                        <a href="{{ url('deleteProjection/'. $p-
>id) }}"><span class="fa fa-remove"></span></a>
                    </div>
                </div>
                <div class="projAdmBorder"></div>
            @endforeach
        </div>
    </div>
</div>
<div id="adminUsers">
    <div id="adminUsersBg">
        <h2>Users</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <div id="admUsersTableDiv">
            <table id="tableUsers">
                <tr>
                    <th>Num.</th>
                    <th>Username</th>
                    <th>Email</th>
                    <th>Password</th>
                    <th>Role</th>
                    <th>Send news</th>
                    <th>Gender</th>
                    <th></th>
                </tr>

```

```

        <?php $i= 0; ?>
        @foreach($users as $u)
            <?php ++$i; ?>
            <tr>
                <td>{{ $i }}.</td>
                <td>{{ $u->username }}</td>
                <td>{{ $u->email }}</td>
                <td>{{ $u->password }}</td>
                <td>{{ $u->roleName }}.</td>
                <td>
                    @if($u->send_via_mail)
                        {{ "Send." }}
                    @else{{ "Don't send." }}
                    @endif
                </td>
                <td>
                    @if($u->genderName == 'f')
                        {{ "Female." }}
                    @else{{ "Male." }}
                    @endif
                </td>
                <td>
                    <div><span class="fa fa-edit" data-id="{{ $u->idUser }}"></span></div>
                    <div><a href="{{ url('deleteUser/' . $u->idUser) }}"><span class="fa fa-remove"></span></a></div>
                </td>
            </tr>
        @endforeach
    </table>
</div>
</div>
</div>
<div id="commentsLinksAdmin">
    <div id="commentsAdmin">
        <h2>Comments</h2>
        <div class="decorationalDiv">
            <span></span>
        </div>
        <div id="commentsHolderAdmin">
            @foreach($comments as $c)
                <div class="adminComment">
                    <div class="comment">
                        <div class="userFaHolder">
                            <span class="fa fa-user"
                                @if($c->gender_id == 1)
                                    {{ "femaleUser " }}
                                @else
                                    {{ "maleUser " }}
                                @endif
                            "></span>
                            <p class="memberSince">Member since</p>
                            <?php
                                $toDate = date_create($c->registered_at);
                                $formattedDate = date_format($toDate,
                                    'd.m.Y');
                            ?>
                            <p
                                class="memberDate">{{ $formattedDate }}</p>

```

```

        </div>
        <div class="commentRight">
            <div class="topPartComment">
                <span class="usernameComment">{{$c-
>username}}</span>

                <?php
                    $toDate = date_create($c->posted_at);
                    $formattedDate = date_format($toDate,
'd.m.Y H:i');
                ?>
                <span
class="commPosted">{{$formattedDate}}</span>
                @if($c->updated_at)
                    <span
class="editedComm">Edited</span>
                @endif
            </div>
            <div class="commentTextDiv">
                <div class="commText">{{$c-
>text}}</div>

                </div>
            </div>
            <span class="fa fa-remove" data-id="{{$c-
>id}}"></span>

        </div>
    </div>
    @endforeach
</div>
</div>
<div class="decorationalAdminDiv"></div>
<div id="linksAdmin">
    <h2>Links</h2>
    <div class="decorationalDiv">
        <span></span>
    </div>
    <button id="insertLink"><span class="fa fa-
plus"></span>New</button>
    <div id="tableLinksDiv">
        <table id="linksTable">
            <tr>
                <th>Num.</th>
                <th>Text</th>
                <th>Path</th>
                <th>In header</th>
                <th>In footer</th>
                <th>Admin menu</th>
                <th></th>
            </tr>
            <?php $i = 0; ?>
            @foreach($links as $l)
                <?php $i++; ?>
                <tr>
                    <td>{{$i}}.</td>
                    <td>{{$l->text}}</td>
                    <td>{{$l->href}}</td>
                    <td>
                        @if($l->in_header)
                            {"Yes."}
                        @else

```

```

                {{ "No." }}
            @endif
        </td>
        <td>
            @if($l->in_footer)
                {{ "Yes." }}
            @else
                {{ "No." }}
            @endif
        </td>
        <td>
            @if($l->admin_link)
                {{ "Yes." }}
            @else
                {{ "No." }}
            @endif
        </td>
        <td>
            <div><span class="fa fa-edit" data-id="{{ $l-
>menu_id }}"></span></div>
            <div><a href="{{ url('deleteLink/' . $l-
>menu_id) }}"><span class="fa fa-remove"></span></a></div>
        </td>
    </tr>
    @endforeach
</table>
</div>
</div>
<div id="activity">
    <h2>Activity</h2>
    <div class="decorationalDiv">
        <span></span>
    </div>
    <div id="activityListHolder">
        <select id="activityList" name="activityList"
class="inputAdminClass">
            <option value="DESC">Newest</option>
            <option value="ASC">Oldest</option>
        </select>
    </div>
    <div id="userActivity">
        <div id="userActivityAbs">
            <table id="activityTable">
                <tr><th>Activity</th><th>Date</th></tr>
                @foreach($activities as $a)
                    <tr>
                        <td>{{ $a->text }}</td>
                        <td>{{ $a->activity_time
}}</td>
                    </tr>
                @endforeach
            </table>
        </div>
    </div>
</div>
<div id="top">

```

```

        <a href="#" target="_top"> <span class="fa fa-arrow-up"></span></a>
    </div>
@endsection

```

Fixed

4.1.12 head.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>MovieBlackout - @yield("title")</title>
    <meta charset="utf-8"/>
    <meta name="description" content="Watch newest movies with your friends and
family, all while enjoying best popcorns you've ever tried.">
    <meta name="keywords" content="Cinema, watch, movies, tickets"/>
    <meta name="author" content="mailto:valerijakoncar20@gmail.com">
    <link rel="shortcut icon" href="{{asset('images/edited/popcorn.ico')}}"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <meta name="language" content="english"/>
    <meta name="copyright" content="MovieBlackout Cinema"/>
    <meta name="robots" content="index, follow"/>
    <link href="https://fonts.googleapis.com/css2?family=Nunito+Sans&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" type="text/css" href="{{ asset('css/style.css') }}">
</head>

```

4.1.13 header.blade.php

```

<body>
    <div id="blurBackground">
        <div id="closeRegFormHolder">
            <div id="closeRegForm"></div>
        </div>
        <div id="registrationHolder">
            <div id="regHeadline">
                <div id="regHeadlineOpacity">
                    <div id="popcornRegForm" class="popcorn"></div>
                    <p id="regP">Registration</p>
                </div>
            </div>
            <form id="registrationForm" action="#" method="POST">
                @csrf
                <table>
                    <tr><td colspan="2"><input type="text" id="regName"
name="regName" placeholder="Username *"/><p class="error"
id="nameError"></p></td></tr>
                    <tr><td colspan="2"><input type="password" id="regPass"
name="regPass" placeholder="Password *"/><p class="error"
id="passError"></p></td></tr>
                    <tr><td colspan="2"><input type="password" id="regPass1"
name="regPass1" placeholder="Type in password again *"/><p class="error"
id="regPass1Error"></p></td></tr>
                    <tr><td><input type="text" id="email" name="email"

```

```

placeholder="E-mail *"/><p class="error" id="mailError"></p></td>
      <td class="marginTd"><input type="text" id="tel"
name="tel" placeholder="Phone (recommended)"/><p class="error"
id="telError"></p></td></tr>
      <tr>
        <td>Gender: *<p class="error" id="genderError"></p></td>
        <td>
          <label class="container" id="firstContainer">Female
            <input type="radio" checked="checked"
name="gender" value="1" id="female"/>
            <span class="checkmark"></span>
          </label>
          <label class="container">Male
            <input type="radio" name="gender" value="2"
id="male">
            <span class="checkmark"></span>
          </label>
        </td>
      </tr>
      <tr><td colspan="2">
        <label class="containerChb">Send news to email
          <input type="checkbox" id="chbMail"
checked="checked" value="send" name="chbMail"/>
          <span class="checkmarkChb"></span>
        </label>
      </td></tr>
      <tr><td colspan="2"><input type="button" id="btnRegistration"
value="Registrare"/></td></tr>
      <tr><td><p class="success"></p></td></tr>
    </table>
  </form>
</div>

</div>

<div id="blurBackgroundPrices">
  <div id="closePricesHolder">
    <div id="closePrices"></div>
  </div>
  <div id="pricesHolder">
    <div id="pricesHeadline">
      <div id="pricesHeadlineOpacity">
        <div id="popcornPrices" class="popcorn"></div>
        <p id="pricesP">Our prices</p>
      </div>
    </div>
    <table id="pricesTable">
      <tr>
        <th>Weekday (Technology)</th>
        <th>Price</th>
      </tr>
      @foreach($prices as $p)
        <tr>
          <td>{{ $p->text }}</td>
          <td>{{ $p->price }} $</td>
        </tr>
      @endforeach
    </table>
  </div>
</div>

```



```

</div>

<div id="blurBackgroundReservation">
    <div id="closeResFormHolder">
        <div id="closeResForm"></div>
    </div>
    <div id="reservationHolder">
        <div id="resHeadline">
            <div id="resHeadlineOpacity">
                <div id="popcornResForm" class="popcorn"></div>
                <p id="resP">Reservation</p>
            </div>
        </div>
        <p id="alertRes"><span class="fa fa-exclamation-
triangle"></span>Please wear musks during ur visit. Safety first!</p>
        @if(session()->has('user'))
            <form id="reservationForm" action="{ { url('/continueReservation')
}} " method="GET">
                {{--
                    @csrf--}}
                    <?php date_default_timezone_set('Europe/Belgrade');?>
                    <?php
                    $todayDate = date("d.m.Y.", strtotime("2020-12-26"));
                    ?>
                    <p id="chooseInfoRes">Please choose your info :</p>
                    <div id="resDdlHolder">
                        <div class="custom-select" id="reservationDateDdlDiv">
                            <select id="reservationDateDdl"
name="reservationDateDdl">
                                <option value="<?=date('Y-m-
d',strtotime($todayDate));?>">Today, <?= date("D d.m.Y.", strtotime(
$todayDate)); ?></option>
                                <option value="<?= date("Y-m-d", strtotime(
$todayDate . "1 day" )); ?>">Tomorrow, <?= date("D d.m.Y.", strtotime( $todayDate
. "1 day" )); ?></option>
                                <?php for($i = 2;$i <= 6;$i++){ ?>
                                    <option value="<?= date("Y-m-d", strtotime(
$todayDate . $i . " day" )); ?>"><?= date("D d.m.Y.", strtotime( $todayDate . $i .
" day" )); ?></option>
                                <?php } ?>
                            </select>
                        </div>
                        <div class="custom-select" id="reservationMovieDdlDiv">
                            <select id="reservationMovieDdl"
name="reservationMovieDdl">
                                <option value="0">Choose movie..</option>
                                @foreach($moviesResDdl as $m)
                                    <option value="{{ $m->id }}">{{ $m-
>name}}</option>
                                @endforeach
                            </select>
                        </div>
                        <div class="custom-select"
id="reservationProjectionDdlDiv">
                            <select id="reservationProjectionDdl"
name="reservationProjectionDdl">
                                <option value="0">Choose projection..</option>
                                </select>
                        </div>

```

```

        </div>
        <div id="chooseNumSeats">Number of seats : <input
type="number" id="numSeats" name="numSeats" min="1" max="10" value="1"></div>
        <div id="continueReservationDiv">
            <span class="fa fa-hand-o-right"></span><input
type="submit" id="continueReservation" value="Next"/>
        </div>
    </form>
    @else
    <p class="notLoggedIn">You are not logged in.</p>
    @endif
</div>

</div>

<div id="blurBackgroundForgetPass">
    <div id="closeForgetPassFormHolder">
        <div id="closeForgetPassForm"></div>
    </div>
    <div id="forgetPassHolder">
        <div id="forgetPassHeadline">
            <div id="forgetPassHeadlineOpacity">
                <div id="popcornForgetPassForm" class="popcorn"></div>
                <p id="forgetPassP">Forgotten password</p>
            </div>
        </div>
        <p id="alertForgetPass"><span class="fa fa-exclamation-
triangle"></span>Confirmation email will be sent to given email address.</p>
        <form id="forgetPassForm" action="{{ url('/forget-password') }}"
method="POST">
            @csrf
            <input type="text" id="forgetPassEmail" name="forgetPassEmail"
placeholder="Your email address.." />
            <p class="error"></p>
            <p class="success"></p>
            <input type="submit" id="forgetPassSubmit"
name="forgetPassSubmit" value="Send" />
        </form>
    </div>
</div>

</div>

<div id="headerWrapper">
    <div id="header">
        <div id="topHeader">
            <div id="logReg">
                <ul id="logRegList">
                    <?php if(session()->has('user')){ ?>
                    <li><a><?= session('user')->username?></a></li>
                    <li><a href="{{ url('/logout') }}"
id="logout">Logout</a></li>
                    <li id="help"><span id="ourPrices"></span><a href="#"
id="help">Our prices</a></li>
                    <?php }else{ ?>
                    <li><a href="#" id="logLink"><span class="fa fa-sign-
in"></span>Log In</a></li>
                    <li><a href="#" id="regLink"><span class="fa fa-

```

```

user"></span>Sign Up</a></li>
        <li><span id="ourPrices"></span><a href="#"
id="help">Our prices</a></li>
        <?php } ?>
    </ul>

    <div id="logFormDiv">
        <div id="closeLogForm">
            <div id="logFormX"></div>
        </div>
        <form method="POST" action="{{ url('/login') }}"
id="logForm">
            @csrf
            <input type="text" name="logUsername" id="logUsername"
placeholder="Username.."/><p class="error" id="logUsernameError"></p>
            <input type="password" name="logPass" id="logPass"
placeholder="Password.."/><p class="error" id="logPassError"></p>
            <input type="submit" name="logIn" id="logIn" value="Log
in" />

            <a href="#" id="resetPass">Forgot password?</a>

            <p class="error" id="logError">
                @isset($errors)
                    @foreach($errors->all() as $error)
                        {{ $error }}<br/>
                    @endforeach
                @endisset
                @if(session()->has('message'))
                    {{ session('message') }}
                @endif
            </p>
        </form>
    </div>
</div>
<a href="{{url('home')}}">
    <div id="logoMessg">
        <div id="logoHeader" class="logo">
            <div id="popcornHeader" class="popcorn"></div>
            <h1>MovieBlackout</h1>
        </div>
        <div id="messageContainer">
            <p class="message">Be kind. Be well. </p>
            <p class="message">Stay safe.</p>
        </div>
    </div>
</a>
<div class="cleaner"></div>
</div>
<div id="menu">
    <div id="menuBtnDiv">
        <button id="menuBtn">
            <i class="fa fa-list"></i>Menu
        </button>
    </div>
    <ul id="menuList">
        @foreach($menuLinks as $link)
            @if($link->in_header && $link->parent_id == NULL)

```

```

        <li>
            <a href="{{ url('.'.$link->href) }}">
                {{ $link->text }}
                @if(count( $link->childrenLinks))
                    <span class="fa fa-angle-
down"></span>
                @endif
            </a>
            @component("partials.menu_children",[
                'children' =>
$link->childrenLinks,
                'menuLinks' =>
$menuLinks
            ])
            @endcomponent
        </li>
    @endif
@endforeach
@if(session()->has("user"))
    @if(session("user")->role_id == 2)
        <li><a href="{{ url('/admin') }}">Admin</a></li>
    @endif
@endif
</ul>
</div>
</div>

```

4.1.14 footer.blade.php

```

<div id="footerContainer">
    <div class="decorationalFooterDiv">
        <span></span>
        <div class="decorationalFooterChild"></div>
        <span></span>
    </div>
    <div id="footer">
        <div id="footerInfoCont">
            <div id="logoSocialsDiv">
                <a href="{{url('home')}}">
                    <div id="footerLogo" class="logo">
                        <div id="popcornFooter" class="popcorn"></div>
                        <h3>MovieBlackout</h3>
                    </div>
                </a>
                <div id="followSocials">
                    <p id="followUs">Follow us on social media<span class="fa
fa-heart-o"></span></p>
                    <ul id="footerSocials">
                        @foreach($socials as $social)
                            <li><a href="{{ $social->href }}"><span
class="{{ $social->icon_class }}"></span></a></li>
                        @endforeach
                    </ul>
                </div>
            </div>
            <div id="footerInfos">
                @foreach($footerHeadlines as $fh)

```



```

        @endif
    </div>
    <div class="commentTextDiv">
        <div class="commText">{{ $c->text }}</div>
    </div>
    <div class="bottomPartComment">
        <span class="fa fa-angle-down showReplies angle" data-id="{{ $c-
>id }}"></span>
        <span class="fa fa-angle-up hideReplies angle" data-id="{{ $c-
>id }}"></span>
        @if(session()->has('user'))
            <span class="reply" data-id="{{ $c->id }}">Reply</span>
            @if(session('user')->id === $c->user_id)

                <span class="fa fa-edit" data-id="{{ $c->id }}"></span>
                <span class="fa fa-trash" data-id="{{ $c->id }}"></span>

            @endif
        @endif
    </div>
</div>
<div class="repliesDiv">

</div>
</div>

```

4.1.16 footer_section.blade.php

```

<div class="footerInfo">
    <h4>{{ $footerHeadline->headline }}</h4>
    <ul>
        @foreach($footerSectionMenu as $fsm)
            @if($footerHeadline->id == $fsm->footer_headline_id)
                <li>
                    <a href="{{ $fsm->href }}">{{ $fsm->text }}</a>
                </li>
            @endif
        @endforeach
    </ul>
</div>

```

4.1.17 menu_children.blade.php

```

@if(count($children)>0)
    <ul class="menuChildrenList notVisible">
        @foreach($children as $child)
            <li>
                <a href="{{ url('.'.$child->href) }}">{{ $child->text }}</a>

                @foreach($menuLinks as $menuLink)
                    @if($menuLink->id == $child->id)
                        @component('partials.menu_children', [
                            'children' => $menuLink->childrenLinks,
                            'menuLinks' => $menuLinks])
                        @endcomponent
                    @endif
                @endforeach
            </li>

```

```

        @endforeach
    </ul>
@endif

```

4.1.18 movie.blade.php

```

<div class="movie">
    <div class="picGeneralMovieInfo">
        <div class="moviePic">
            <a href="{{ url('movie/'.$movie->idMovie) }}">
                @if(strpos($movie->path, "cloudinary"))
                    alt }}" width="250" height="350"/>
                @else
                    alt }}" width="250" height="350"/>
                @endif
            </a>
        </div>
        <div class="generalMovieInfo">
            <a href="{{ url('movie/'.$movie->idMovie) }}">
                <h3>{{ $movie->movieName }}<span class="ageLimit">{{ $movie->age_limit }}</span></h3>
            </a>
            <div class="movieGrade">
                <p class="numberMovieGrade">{{ $movie->grade }}</p>
                @component("partials.movie_stars",["movie" => $movie])
                    @endcomponent
                <?php if(session()->has('user') && session("user")->role_id == 1){ ?>
                    <a href="#" class="vote">Rate?</a>
                    <div class="voteDiv">
                        <div class="giveVote">
                            <div class="faCloseRateCloud"><span class="fa fa-close"></span></div>
                            <p>{{ $movie->movieName }}</p>
                            <ul class="giveVoteList">
                                @for($i = 0; $i < 10; $i++)
                                    <li><span class="fa fa-star-o" data-index="{{ $i }}"></span></li>
                                @endfor
                            </ul>
                            <span class="userChosenGrade">0.0</span>
                            <input type="button" value="Rate" class="confirmVote" data-movid="{{ $movie->idMovie }}" />
                        </div>
                    </div>
                    <?php } ?>
                </div>
            <p class="movieDesc">{{ $movie->description }}</p>
            <p>
                <?php $i=0; ?>
                @foreach($movie->genres as $genre)
                    <?php $i++; ?>
                    {{ $genre->genre }}
                    @if(count($movie->genres) != $i)
                        {{ "," . " " }}
                    @endif
                @endforeach
                <span class="fa fa-clock-o"></span> {{ $movie->running_time }}
            </p>
        </div>
    </div>

```

```

min
        </p>
        <p class="movieDate">In cinemas from:
            <?php
                $toDate = date_create($movie->in_cinemas_from);
                $formattedDate = date_format($toDate, 'd.m.Y.');
```

4.1.19 movie_starts.blade.php

```

<ul>
    <?php
        $gradeNumber = intval($movie->grade);
        $firstDecimal = intval(substr($movie->grade,2,1));
    //    dd($firstDecimal);
    for($i=0;$i<$gradeNumber;$i++){
        <?php
        }
        if(($firstDecimal != 0) && ($firstDecimal >= 3)){
            if($firstDecimal <= 7){
                $i++;
            }
            <li><span class="fa fa-star-half-o"></span></li>
        }
        }else{
            $i++;
        }
        <li><span class="fa fa-star"></span></li>
    }
    }
    for($i;$i < 10; $i++){
        <?php
        }
        <li><span class="fa fa-star-o"></span></li>
    }
    }
    </ul>
```


4.1.20 projection.blade.php

```
<div class="movieProjection"
  @if(!$p->reservation_available) || $p->projectionAlreadyStartedEnded)
    {{ " disabledReservation" }}
  @endif
">
  @if(session()->has('user'))
    @if($p->reservation_available && (!$p->projectionAlreadyStartedEnded))
      <a href="{{ url('/continueReservation/' . $p->id) }}" data-id="{{ $p->id }}" class="projectionLink">
    @endif
  @endif
  <p class="movieTime">
    <?php
      $toDate = date_create($p->starts_at);
      $formattedDate = date_format($toDate, 'H:i');
    ?>
    {{ $formattedDate }}
    @if(!$p->reservation_available) || $p->projectionAlreadyStartedEnded)
      <span class="fa fa-ticket redTicket"></span>
    @else
      <span class="fa fa-ticket greenTicket"></span>
    @endif
  </p>
  <p class="movieHall">{{ $p->name }}</p>
  <p class="movieTechn">{{ $p->technologyName }}</p>
  @if(session()->has('user'))
    @if($p->reservation_available && (!$p->projectionAlreadyStartedEnded))</a>@endif
  @endif
</div>
```

4.1.21 template.blade.php

```
@include("fixed.head")
@include("fixed.header")
@yield("mainContent")
@include("fixed.footer")
```

Admin

4.1.22 admin_template.blade.php

```
@include("admin.fixed.head")
@include("admin.fixed.header")
@yield("mainContent")
@include("admin.fixed.footer")
```

Admin/fixed

4.1.23 head.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<title>MovieBlackout - @yield("title")</title>
<meta charset="utf-8"/>
<link rel="shortcut icon" href="{{asset('images/edited/popcorn.ico')}}"/>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" type="text/css" href="{{ asset("css/style.css") }}" />
</head>

```

4.1.24 header.blade.php

```

<body>
<div class="blurBackgroundsAdmin" id="blurBgUpdateMovie">
    <div id="closeUpdateMovieFormHolder">
        <div id="closeUpdateMovieForm"></div>
    </div>
    <div id="updateMovieHolder">
        <p class="adminFormHeadline">Update Movie</p>
        <form id="updateMovieForm" action="{{url('updateMovie')}}" method="POST"
enctype="multipart/form-data">
            @csrf
            <input type="hidden" id="movieIdHidden" name="movieIdHidden"/>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Movie name</label>
                <div class="labelInputHolderMultiple">
                    <input type="text" placeholder="Movie Name.."
class="inputAdminClass" id="nameMovUpd" name="nameMovUpd"/>
                    <label class="inputAdminLabel inputAdminLabelSecond">Running
time</label>
                    <input type="text" placeholder="Running time.."
class="inputAdminClass" id="timeMovUpd" name="timeMovUpd"/>
                </div>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Movie description</label>
                <textarea id="taShortDesc" name="taShortDesc"
class="inputAdminClass"></textarea>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Detailed description</label>
                <textarea id="taDetailedDesc" name="taDetailedDesc"
class="inputAdminClass"></textarea>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Distributer</label>
                <div class="labelInputHolderMultiple">
                    {{--
class="inputAdminClass" id="distributerMovUpd"/>--}}
                    <select id="distributerMovUpd" name="distributerMovUpd"
class="inputAdminClass">
                        <option value="0">Choose distributer.</option>
                        @foreach($distributers as $d)
                            <option value="{{ $d->id }}" {{ $d->name }}</option>
                        @endforeach
                    </select>
                    <label class="inputAdminLabel
inputAdminLabelSecond">Director</label>
                    <select id="directorMovUpd" name="directorMovUpd"
class="inputAdminClass">

```

```

        <option value="0">Choose director..</option>
        @foreach($directors as $d)
        <option value="{{ $d->id }}">{{ $d->name }}</option>
        @endforeach
    </select>
    <label class="inputAdminLabel
inputAdminLabelSecond">Technology</label>
    <select id="technologiesMovUpd" name="technologiesMovUpd"
class="inputAdminClass">
        @foreach($technologies as $t)
        <option value="{{ $t->id }}">{{ $t->name }}</option>
        @endforeach
        <option value="3">All technologies.</option>
    </select>
</div>
</div>
<div class="labelInputHolder">
    <label class="inputAdminLabel">Country</label>
    <div class="labelInputHolderMultiple">
        <select id="countryMovUpd" name="countryMovUpd"
class="inputAdminClass">
            <option value="0">Choose country..</option>
            @foreach($countries as $c)
            <option value="{{ $c->id }}">{{ $c->name }}</option>
            @endforeach
        </select>
        <label class="inputAdminLabel inputAdminLabelSecond">In
cinemas</label>
        <input type="date" class="inputAdminClass" id="dateMovUpd"
name="dateMovUpd"/>
    </div>
</div>
<div class="labelInputHolder">
    <label class="inputAdminLabel">Actors</label>
    <div class="selectAdminMovHolder">
        <select class="inputAdminClass" id="actorsMovUpd"
name="actorsMovUpd">
            <option value="0">Choose actor..</option>
            @foreach($actors as $a)
            <option value="{{ $a->id }}">{{ $a->name }}</option>
            @endforeach
        </select>
        <div id="actorsListMovUpd">
        </div>
    </div>
</div>
<div class="labelInputHolder">
    <label class="inputAdminLabel">Genres</label>
    <div class="selectAdminMovHolder">
        <select class="inputAdminClass" id="genresMovUpd"
name="genresMovUpd">
            <option value="0">Choose genre..</option>
            @foreach($genres as $g)
            <option value="{{ $g->id }}">{{ $g->genre }}</option>
            @endforeach
        </select>
        <div id="genresListMovUpd">
        </div>
    </div>
</div>

```

```

        </div>
        <div class="labelInputHolder">
            <label class="inputAdminLabel">Age limit</label>
            <div class="labelInputHolderMultiple">
                <input type="text" class="inputAdminClass" id="ageMovUpd"
name="ageMovUpd" placeholder="Age limit.."/>
                <label class="inputAdminLabel inputAdminLabelSecond">Cooming
soon</label>
                <select class="inputAdminClass" id="csMovUpd" name="csMovUpd">
                    <option value="0">NO.</option>
                    <option value="1" selected="true">YES.</option>
                </select>
                <label class="inputAdminLabel inputAdminLabelSecond">Active
Slider</label>
                <select class="inputAdminClass" id="activeSliderMovUpd"
name="activeSliderMovUpd">
                    <option value="0" selected="true">NO.</option>
                    <option value="1">YES.</option>
                </select>
            </div>
        </div>
        <div class="labelInputHolder">
            <label class="inputAdminLabel">Movie poster</label>
            <div class="labelInputHolderMultiple">
                <input type="file" class="inputAdminClass" id="picMovUpd"
name="picMovUpd"/>
                <label class="inputAdminLabel inputAdminLabelSecond">Movie
bg</label>
                <input type="file" class="inputAdminClass" id="picMovBgUpd"
name="picMovBgUpd"/>
                <label class="inputAdminLabel inputAdminLabelSecond">Slider
pic</label>
                <input type="file" class="inputAdminClass"
id="picSliderMovUpd" name="picSliderMovUpd"/>
            </div>
        </div>
        <input type="submit" value="Update" id="updMov"/>
        <p class="errors error"></p>
        @isset($errors)
        <p class="error">
            @foreach($errors->all() as $err)
                {{ $err }}
            @endforeach
        </p>
        @endisset
        @if(session()->has("error"))
        <p class="error">{{session("error")}}</p>
        @endif
        @if(session()->has("success"))
        <p class="success">{{session("success")}}</p>
        @endif
    </form>
</div>

<div class="blurBackgroundsAdmin" id="blurBgInsertMovie">
    <div id="closeInsertMovieFormHolder">
        <div id="closeInsertMovieForm"></div>
    </div>

```

```

    </div>
    <div id="insertMovieHolder">
        <p class="adminFormHeadline">Insert Movie</p>
        <form id="insertMovieForm" action="{{url('insertMovie')}}" method="POST"
enctype="multipart/form-data">
            @csrf
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Movie name</label>
                <div class="labelInputHolderMultiple">
                    <input type="text" placeholder="Movie Name.."
class="inputAdminClass" id="nameMovIns" name="nameMovIns"/>
                    <label class="inputAdminLabel inputAdminLabelSecond">Running
time</label>
                    <input type="text" placeholder="Running time.."
class="inputAdminClass" id="timeMovIns" name="timeMovIns"/>
                </div>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Movie description</label>
                <textarea id="taShortDescIns" name="taShortDescIns"
class="inputAdminClass"></textarea>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Detailed description</label>
                <textarea id="taDetailedDescIns" name="taDetailedDescIns"
class="inputAdminClass"></textarea>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Distributer</label>
                <div class="labelInputHolderMultiple">
                    {{-- <input type="text"
placeholder="Distrubuter.." class="inputAdminClass" id="distributerMovUpd"/>--}}
                    <select id="distributerMovIns" name="distributerMovIns"
class="inputAdminClass">
                        <option value="0">Choose distributer..</option>
                        @foreach($distributers as $d)
                            <option value="{{ $d->id }}">{{ $d->name }}</option>
                        @endforeach
                    </select>
                    <label class="inputAdminLabel
inputAdminLabelSecond">Director</label>
                    <select id="directorMovIns" name="directorMovIns"
class="inputAdminClass">
                        <option value="0">Choose director..</option>
                        @foreach($directors as $d)
                            <option value="{{ $d->id }}">{{ $d->name }}</option>
                        @endforeach
                    </select>
                    <label class="inputAdminLabel
inputAdminLabelSecond">Technology</label>
                    <select id="technologiesMovIns" name="technologiesMovIns"
class="inputAdminClass">
                        @foreach($technologies as $t)
                            <option value="{{ $t->id }}">{{ $t->name }}</option>
                        @endforeach
                        <option value="3">All technologies.</option>
                    </select>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="labelInputHolder">
            <label class="inputAdminLabel">Country</label>
            <div class="labelInputHolderMultiple">
                <select id="countryMovIns" name="countryMovIns"
class="inputAdminClass">
                    <option value="0">Choose country..</option>
                    @foreach($countries as $c)
                        <option value="{{ $c->id }}" {{ $c->name }}</option>
                    @endforeach
                </select>
                <label class="inputAdminLabel inputAdminLabelSecond">In
cinemas</label>
                <input type="date" class="inputAdminClass" id="dateMovIns"
name="dateMovIns"/>
            </div>
        </div>
        <div class="labelInputHolder">
            <label class="inputAdminLabel">Actors</label>
            <div class="selectAdminMovHolder">
                <select class="inputAdminClass" id="actorsMovIns"
name="actorsMovIns">
                    <option value="0">Choose actor..</option>
                    @foreach($actors as $a)
                        <option value="{{ $a->id }}" {{ $a->name }}</option>
                    @endforeach
                </select>
                <div id="actorsListMovIns">
                </div>
            </div>
        </div>
        <div class="labelInputHolder">
            <label class="inputAdminLabel">Genres</label>
            <div class="selectAdminMovHolder">
                <select class="inputAdminClass" id="genresMovIns"
name="genresMovIns">
                    <option value="0">Choose genre..</option>
                    @foreach($genres as $g)
                        <option value="{{ $g->id }}" {{ $g->genre }}</option>
                    @endforeach
                </select>
                <div id="genresListMovIns">
                </div>
            </div>
        </div>
        <div class="labelInputHolder">
            <label class="inputAdminLabel">Age limit</label>
            <div class="labelInputHolderMultiple">
                <input type="text" class="inputAdminClass" id="ageMovIns"
name="ageMovIns" placeholder="Age limit.."/>
                <label class="inputAdminLabel inputAdminLabelSecond">Cooming
soon</label>
                <select class="inputAdminClass" id="csMovIns" name="csMovIns">
                    <option value="0">NO.</option>
                    <option value="1" selected="true">YES.</option>
                </select>
                <label class="inputAdminLabel inputAdminLabelSecond">Active
Slider</label>
                <select class="inputAdminClass" id="activeSliderMovIns"
name="activeSliderMovIns">

```

```

        <option value="0" selected="true">NO.</option>
        <option value="1">YES.</option>
    </select>
</div>
</div>
<div class="labelInputHolder">
    <label class="inputAdminLabel">Movie poster</label>
    <div class="labelInputHolderMultiple">
        <input type="file" class="inputAdminClass" id="picMovIns"
name="picMovIns"/>
        <label class="inputAdminLabel inputAdminLabelSecond">Movie
bg</label>
        <input type="file" class="inputAdminClass" id="picMovBgIns"
name="picMovBgIns"/>
        <label class="inputAdminLabel inputAdminLabelSecond">Slider
pic</label>
        <input type="file" class="inputAdminClass"
id="picSliderMovIns" name="picSliderMovIns"/>
    </div>
</div>
<input type="submit" value="Insert" id="insMov"/>
<p class="errors error"></p>
@isset($errors)
    <p class="error">
        @foreach($errors->all() as $err)
            {{ $err }}
        @endforeach
    </p>
@endisset
@if(session()->has("errorIns"))
    <p class="error">{{session("errorIns")}}</p>
@endif
@if(session()->has("successIns"))
    <p class="success">{{session("successIns")}}</p>
@endif
</form>
</div>
</div>

<div class="blurBackgroundsAdmin" id="blurUpdateProj">
    <div id="closeUpdateProjFormHolder">
        <div id="closeUpdateProjForm"></div>
    </div>
    <div id="updateProjHolder">
        <p class="adminFormHeadline">Update projection</p>
        <form id="updateProjForm" action="{{url('updateProjection')}}"
method="POST">
            @csrf
            <input type="hidden" id="idProjHidden" name="idProjHidden"/>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Movie</label>
                <select id="movieProjectionUpdate" name="movieProjectionUpdate"
class="inputAdminClass">
                    <option value="0">Choose movie..</option>
                    @foreach($allMovies as $m)
                        <option value="{{ $m->id }}">{{ $m->name }}</option>
                    @endforeach
                </select>
            </div>
        </form>
    </div>
</div>

```

```

        </select>
    </div>
    <div class="labelInputHolder">
        <label class="inputAdminLabel">Projection date</label>
        <input type="date" id="dateProjectionUpdate"
name="dateProjectionUpdate" class="inputAdminClass"/>
        <label class="inputAdminLabel inputAdminLabelSecond">Time</label>
        <input type="text" id="timeProjectionUpdate"
name="timeProjectionUpdate" class="inputAdminClass"/>
    </div>
    <div class="labelInputHolder">
        <label class="inputAdminLabel">Technology</label>
        <select id="technProjectionUpdate" name="technProjectionUpdate"
class="inputAdminClass">
            <option value="0"></option>
        </select>
        <label class="inputAdminLabel
inputAdminLabelSecond">Reservation</label>
        <select id="resProjectionUpdate" name="resProjectionUpdate"
class="inputAdminClass">
            <option value="0">Not available.</option>
            <option value="1">Available.</option>
        </select>
    </div>
    <div class="labelInputHolder">
        <label class="inputAdminLabel">Hall</label>
        <select id="hallProjectionUpdate" name="hallProjectionUpdate"
class="inputAdminClass">
            <option value="0">Choose hall..</option>
            @foreach($halls as $h)
                <option value="{{ $h->id }}">{{ $h->name }}</option>
            @endforeach
        </select>
    </div>
    <input type="submit" value="Update" id="updProj"/>
    <p class="errors error"></p>
    @isset($errors)
        <p class="error">
            @foreach($errors->all() as $err)
                {{ $err }}
            @endforeach
        </p>
    @endisset
    @if(session()->has("errorUpdProj"))
        <p class="error">{{session("errorUpdProj")}}</p>
    @endif
    @if(session()->has("successUpdProj"))
        <p class="success">{{session("successUpdProj")}}</p>
    @endif
</form>
</div>
</div>

<div class="blurBackgroundsAdmin" id="blurBgUpdateUser">
    <div id="closeUpdateUserFormHolder">
        <div id="closeUpdateUserForm"></div>
    </div>
    <div id="updateUserHolder">
        <p class="adminFormHeadline">Update User</p>
    </div>

```



```

        <form id="updateUserForm" action="#" method="POST">
            @csrf
            <input type="hidden" id="userIdHidden" name="userIdHidden"/>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Username</label>
                <div class="labelInputHolderMultiple">
                    <input type="text" placeholder="Username.."
class="inputAdminClass" id="usernameUpd" name="usernameUpd"/>
                    <label class="inputAdminLabel
inputAdminLabelSecond">Role</label>
                    <select id="roleUserUpd" name="roleUserUpd"
class="inputAdminClass">
                        @foreach($roles as $r)
                            <option value="{{ $r->id }}" {{ $r->role }}</option>
                        @endforeach
                    </select>
                </div>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Password</label>
                <input type="password" placeholder="New password.."
class="inputAdminClass" id="passwordUpd" name="passwordUpd"/>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Email</label>
                <input type="text" placeholder="Email.." class="inputAdminClass"
id="emailUserUpd" name="emailUserUpd"/>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Gender</label>
                <div class="labelInputHolderMultiple">
                    <select id="genderUserUpd" name="genderUserUpd"
class="inputAdminClass">
                        @foreach($genders as $g)
                            <option value="{{ $g->id }}">
                                @if($g->name == "f")
                                    {{ "Female." }}
                                @else
                                    {{ "Male." }}
                                @endif
                            </option>
                        @endforeach
                    </select>
                    <label class="inputAdminLabel inputAdminLabelSecond">Send
news</label>
                    <select id="newsUserUpd" name="newsUserUpd"
class="inputAdminClass">
                        <option value="0">Don't send.</option>
                        <option value="1">Send news.</option>
                    </select>
                </div>
            </div>
            <input type="submit" value="Update" id="updUser"/>
            <p class="errors error"></p>
            <p class="success"></p>
        </form>
    </div>
</div>

```

```

<div class="blurBackgroundsAdmin" id="blurBgUpdateLink">
  <div id="closeUpdateLinkFormHolder">
    <div id="closeUpdateLinkForm"></div>
  </div>
  <div id="updateLinkHolder">
    <p class="adminFormHeadline">Update Link</p>
    <form id="updateLinkForm" action="#" method="POST">
      @csrf
      <input type="hidden" id="linkIdHidden" name="linkIdHidden"/>
      <div class="labelInputHolder">
        <label class="inputAdminLabel">Text</label>
        <div class="labelInputHolderMultiple">
          <input type="text" placeholder="Link text.."
class="inputAdminClass" id="textLinkUpd" name="textLinkUpd"/>
          <label class="inputAdminLabel
inputAdminLabelSecond">Path</label>
          <input type="text" placeholder="Link path.."
class="inputAdminClass" id="pathLinkUpd" name="pathLinkUpd"/>
        </div>
      </div>
      <div class="labelInputHolder">
        <label class="inputAdminLabel">In header</label>
        <div class="labelInputHolderMultiple">
          <select id="headerLinkUpd" name="headerLinkUpd"
class="inputAdminClass">
            <option value="0">No.</option>
            <option value="1">Yes.</option>
          </select>
          <label class="inputAdminLabel inputAdminLabelSecond">In
footer</label>
          <select id="footerLinkUpd" name="footerLinkUpd"
class="inputAdminClass">
            <option value="0">No.</option>
            <option value="1">Yes.</option>
          </select>
        </div>
      </div>
      <div class="labelInputHolder">
        <label class="inputAdminLabel">Parent link</label>
        <div class="labelInputHolderMultiple">
          <select id="parentLinkUpd" name="parentLinkUpd"
class="inputAdminClass">
            <option value="0">None.</option>
            @foreach($links as $l)
              <option value="{{ $l->menu_id }}">{{ $l->text }}</option>
            @endforeach
          </select>
          <label class="inputAdminLabel inputAdminLabelSecond">Admin
menu</label>
          <div class="linkChbHolder">
            <input type="checkbox" id="adminLinkUpd"
name="adminLinkUpd"/>
          </div>
        </div>
      </div>
      <input type="submit" value="Update" id="updLink"/>
      <p class="errors error"></p>
    </form>
  </div>
</div>

```

```

        <p class="success"></p>
    </form>
</div>

</div>

<div class="blurBackgroundsAdmin" id="blurBgInsertLink">
    <div id="closeInsertLinkFormHolder">
        <div id="closeInsertLinkForm"></div>
    </div>
    <div id="insertLinkHolder">
        <p class="adminFormHeadline">Insert Link</p>
        <form id="insertLinkForm" action="#" method="POST">
            @csrf
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Text</label>
                <div class="labelInputHolderMultiple">
                    <input type="text" placeholder="Link text.."
class="inputAdminClass" id="textLinkIns" name="textLinkIns"/>
                    <label class="inputAdminLabel
inputAdminLabelSecond">Path</label>
                    <input type="text" placeholder="Link path.."
class="inputAdminClass" id="pathLinkIns" name="pathLinkIns"/>
                </div>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">In header</label>
                <div class="labelInputHolderMultiple">
                    <select id="headerLinkIns" name="headerLinkIns"
class="inputAdminClass">
                        <option value="0">No.</option>
                        <option value="1">Yes.</option>
                    </select>
                    <label class="inputAdminLabel inputAdminLabelSecond">In
footer</label>
                    <select id="footerLinkIns" name="footerLinkIns"
class="inputAdminClass">
                        <option value="0">No.</option>
                        <option value="1">Yes.</option>
                    </select>
                </div>
            </div>
            <div class="labelInputHolder">
                <label class="inputAdminLabel">Parent link</label>
                <div class="labelInputHolderMultiple">
                    <select id="parentLinkIns" name="parentLinkIns"
class="inputAdminClass">
                        <option value="0">None.</option>
                        @foreach($links as $l)
                            <option value="{{ $l->menu_id }}">{{ $l->text }}</option>
                        @endforeach
                    </select>
                    <label class="inputAdminLabel inputAdminLabelSecond">Admin
menu</label>
                    <div class="linkChbHolder">
                        <input type="checkbox" id="adminLinkIns"
name="adminLinkIns" selected="false"/>
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

```

```

        </div>
        <input type="submit" value="Insert" id="insLink"/>
        <p class="errors error"></p>
        <p class="success"></p>
    </form>
</div>
</div>

<div id="adminHeader">
    <div id="adminHeaderHolder">
        <div id="topAdminHeader">
            <div id="rightAdminHeaderDiv">
                <p id="helloAdmin">Hello
                    @if(session()->has('user'))
                        @if(session('user')->role_id == 2)
                            {{ session('user')->username."!"}}
                        @endif
                    @endif
                    <span class="fa fa-heart-o"></span>
                    <a href="{{ url("/logout") }}" id="logoutlink">Logout</a>
                </div>
                <div id="logoMessg">
                    <a href="{{ url('admin') }}">
                        <div id="logoHeader" class="logo">
                            <div id="popcornHeader" class="popcorn"></div>
                            <h1>MovieBlackout</h1>
                        </div>
                    </a>
                </div>
                <div class="cleaner"></div>
            </div>
            <div id="menuAdminDiv">
                <div id="menuBtnDivAdm">
                    <button id="menuBtnAdm">
                        <i class="fa fa-list"></i>Menu
                    </button>
                </div>
                <ul id="menuAdmin">
                    @foreach($menu as $m)
                        <li><a href="{{ url('.'.$m->href) }}">{{ $m->text}}</a></li>
                    @endforeach
                </ul>
            </div>
        </div>
    </div>
</div>

```

4.1.25 footer.blade.php

```

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="{{ asset('js/main.js') }}" type="text/javascript"></script>
<script type="text/javascript">
    var urlPics = '{{ URL::asset('/images/') }}';
</script>

```

Admin/partial

4.1.26 movie.blade.php

```
<div class="movieAdmin">
    <div class="moviePicAdmin movieInfoAdmin">
        @if(strpos($m->path, "cloudinary"))
            alt }}" />
        @else
            alt }}" />
        @endif
    {{--
        width="120" height="180"--}}
    <div class="underImgMovieAdm">
        <p class="movieDateAdmin">Date aired:
            <?php
                $toDate = date_create($m->in_cinemas_from);
                $formattedDate = date_format($toDate, 'd.m.Y. ');
            ?>
            <span>{{ $formattedDate }}</span>
        </p>
        <p>
            <?php $i=0; ?>
            @foreach($m->genres as $genre)
                <?php $i++; ?>
                {{ $genre->genre }}
                @if(count($m->genres) != $i)
                    {{ "," }}
                @endif
            @endforeach
            | <span class="fa fa-clock-o"></span> {{ $m->running_time }} min
        </p>
    </div>
</div>
<div class="movieNameAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Movie name</p>
    {{ $m->movieName }}
    <p class="headlineAdmMovie headlineAdmActors">Actors</p>
    <p class="actorsAdmMovie">
        @for($i=0; $i<count($m->actors); $i++)
            @if($i === count($m->actors)-1)
                {{ $m->actors[$i]->name }}
            @else
                {{ $m->actors[$i]->name . ", " }}
            @endif
        @endfor
    </p>
</div>
<div class="movieRatingAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Rating</p>
    <p class="numberMovieGrade">{{ $m->grade }}</p>
    @component("partials.movie_stars", ["movie" => $m])
    @endcomponent
</div>
<div class="movieDirectorAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Director</p>
    {{ $m->dirName }}
</div>
<div class="movieDescAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Description</p>
```

```

        <p class="admDescP">{{$m->description}}</p>
    </div>
    <div class="movieInfoAdmin btnsAdminMovies">
        <div><input type="button" value="Update" class="btnUpdateMov" data-
id="{{$m->idMovie}}"/></div>
        <div><a href="{{url('deleteMovie/'. $m->idMovie)}}" data-id="{{$m-
>idMovie}}" class="btnDeleteMov">Delete</a></div>
    </div>
</div>

```

4.2 Controllers

4.2.1 AdminController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Activity;
use App\Models\Actor;
use App\Models\Comment;
use App\Models\Country;
use App\Models\Director;
use App\Models\Distributer;
use App\Models\Gender;
use App\Models\Genre;
use App\Models\Hall;
use App\Models\MenuModel;
use App\Models\Movie;
use App\Models\Projection;
use App\Models\Role;
use App\Models\Technology;
use App\Models\User;
use Illuminate\Http\Request;

class AdminController extends Controller
{
    private $menuModel;
    private $movieModel;
    private $genreModel;
    private $actorModel;
    private $distributerModel;
    private $directorModel;
    private $countryModel;
    private $hallModel;
    private $projectionModel;
    private $technologyModel;
    private $userModel;
    private $genderModel;
    private $roleModel;
    private $commentModel;
    private $activityModel;
    private $data = [];

    public function __construct()
    {
        $this->menuModel = new MenuModel();
        $this->movieModel = new Movie();
        $this->genreModel = new Genre();
    }
}

```

```

        $this->actorModel = new Actor();
        $this->distributerModel = new Distributer();
        $this->directorModel = new Director();
        $this->countryModel = new Country();
        $this->hallModel = new Hall();
        $this->projectionModel = new Projection();
        $this->technologyModel = new Technology();
        $this->userModel = new User();
        $this->genderModel = new Gender();
        $this->roleModel = new Role();
        $this->commentModel = new Comment();
        $this->activityModel = new Activity();
        $this->data["menu"] = $this->menuModel->getMenuLinksAdmin();
        $this->data["movies"] = $this->movieModel->getMoviesAdmin(0);
        $this->data["allMovies"] = $this->movieModel->getAllMoviesAdmin();
        $this->data["pages"] = $this->movieModel->getAdminPagationCount();
        $this->data["genres"] = $this->genreModel->getGenres();
        $this->data["actors"] = $this->actorModel->getActors();
        $this->data["distributers"] = $this->distributerModel->getDistributers();
        $this->data["directors"] = $this->directorModel->getDirectors();
        $this->data["countries"] = $this->countryModel->getCountries();
        $this->data["halls"] = $this->hallModel->getHalls();
        $this->data["projections"] = $this->projectionModel->getProjections();
        $this->data["technologies"] = $this->technologyModel->getTechnologies();
        $this->data["users"] = $this->userModel->getUsers();
        $this->data["genders"] = $this->genderModel->getGenders();
        $this->data["roles"] = $this->roleModel->getRoles();
        $this->data["comments"] = $this->commentModel->getAllComments();
        $this->data["links"] = $this->menuModel->getAllLinks();
        $this->data["activities"] = $this->activityModel->print();
    }

    public function admin(){
        return view("admin.pages.admin", $this->data);
    }
}

```

4.2.2 CommentController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Activity;
use App\Models\Comment;
use Illuminate\Http\Request;

class CommentController extends Controller
{
    private $commentModel;
    private $activityModel;

    public function __construct()
    {
        $this->commentModel = new Comment();
        $this->activityModel = new Activity();
    }

    public function getRepliesOfComment($parent_id){
        $comments = $this->commentModel->getRepliesOfComment($parent_id);
        $loggedUserId = 0;
    }
}

```

```

        if(session()->has('user')) {
            $loggedUserId = session()->get('user')->id;
        }
        return response()->json(["replies" => $comments, "loggedUserId" =>
$loggedUserId], 200);
    }

    public function viewMoreComments($id, $offset){
        $comments = $this->commentModel->viewMoreComments($id, $offset);
        $loggedUserId = 0;
        if(session()->has('user')) {
            $loggedUserId = session()->get('user')->id;
        }
        return response()->json(["comments" => $comments, "loggedUserId" =>
$loggedUserId], 200);
    }

    public function deleteComment($id){
        if(session()->get("user")->role_id == 2){
            $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
deleted comment with id: ". $id . "\t";
            $userId = session()->get('user')->id;
            $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        }
        try{
            $this->commentModel->deleteComment($id);
            return response()->json("", 204);
        }catch(\PDOException $ex){
            return response()->json("", 500);
        }
    }

    public function updateComment(Request $request){
        $modifiedCommText = $request->input('newComment');
        $commentId = $request->input('commentId');
        $dateTime = $this->getDateTime();
        $data = $this->commentModel->updateComment($modifiedCommText, $commentId,
$dateTime);
        $code = $data["code"];
        return response()->json("", $code);
    }

    public function postComment(Request $request){
        $comment = $request->input("comment");
        $idMovie = $request->input("idMovie");
        $idUser = $request->session()->get('user')->id;
        $dateTime = $this->getDateTime();
        if($request->has("parent_id")){
            $parent_id = $request->input("parent_id");
            $data = $this->commentModel->postComment($comment, $idMovie, $idUser,
$dateTime, $parent_id);
        }else{
            $data = $this->commentModel->postComment($comment, $idMovie, $idUser,
$dateTime);
        }
        $code = $data["code"];
        $loggedUserId = $request->session()->get('user')->id;
        $comment = $data["newComment"];
    }

```



```

        return response()->json(["newComment" => $comment, "loggedUserId" =>
$loggedUserId], $code);
    }

    public function getDateTime(){
        date_default_timezone_set('Europe/Belgrade');

        return date("Y-m-d H:i:s");
    }
}

```

4.2.3 FrontController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\AboutCinema;
use App\Models\Author;
use App\Models\MenuModel;
use App\Models\Movie;
use App\Models\PageSectionModel;
use App\Models\Price;
use Illuminate\Http\Request;

class FrontController extends Controller
{
    protected $data = [];
    protected $menuModel;
    protected $pageSecModel;
    private $moviesModel;
    private $pricesModel;

    public function __construct()
    {
        $this->menuModel = new MenuModel();
        $this->pageSecModel = new PageSectionModel();
        $this->moviesModel = new Movie();
        $this->pricesModel = new Price();

        $this->data['menuLinks'] = $this->menuModel->getMenuLinks();
        $dataArray = $this->getDate();
        $this->data['moviesResDdl'] = $this->moviesModel-
>getMoviesDdl($dataArray);
        $this->data['footerHeadlines'] = $this->pageSecModel-
>getPageSection('footer_headline');
        $this->data['socials'] = $this->pageSecModel-
>getPageSection('social_network');
        $this->data['prices'] = $this->pricesModel->getPrices();
    }

    public function membership(){
        return view("pages.membership", $this->data);
    }

    public function aboutCinema(){
        $aboutCinemaModel = new AboutCinema();
        $text = $aboutCinemaModel->getAboutCinemaText()->text;
    }
}

```

```

        $this->data['text'] = $text;
        return view("pages.about_cinema", $this->data);
    }

    public function aboutAuthor(){
        $authorModel = new Author();
        $text = $authorModel->getAuthorInfo()->text;
        $this->data['author'] = $text;
        return view("pages.about_author", $this->data);
    }

    public function websiteAdmin(){
        return view("pages.contact_admin", $this->data);
    }

    public function contactCinema(){
        return view("pages.contact_cinema", $this->data);
    }

    public function getDate($date = 0){
        date_default_timezone_set('Europe/Belgrade');

        if($date){
            $todayDate = $date;
            $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
            $todayDate = date('Y-m-d', strtotime($todayDate));
        }else{
            $todayDate = date("Y-m-d", strtotime("2020-12-26"));
            $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
        }
        $minus1Day = $minus1Day . " 23:59:59";
        $timeNow = " 15:00:00";
        $dateTimeNow = $todayDate . $timeNow;

        $dateArray = [ "plus1Day" => $plus1Day, "minus1Day" => $minus1Day,
"todayDate" => $todayDate, "dateTimeNow" => $dateTimeNow];
        return $dateArray;
    }
}

```

4.2.4 HomeController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Genre;
use App\Models\Movie;
use App\Models\Technology;
use Illuminate\Http\Request;

class HomeController extends FrontController
{
    private $moviesModel;
    private $genreModel;
    private $technologyModel;
}

```

```

    public function __construct()
    {
        parent::__construct();
        $this->moviesModel = new Movie();
        $this->genreModel = new Genre();
        $this->technologyModel = new Technology();
    }

    public function home(){
        $this->data['movies'] = $this->moviesModel->getMovies(parent::getDate());
        $this->data['genres'] = $this->genreModel->getGenres();
        $this->data['technologies'] = $this->technologyModel->getTechnologies();
        $this->data['pagination'] = $this->moviesModel->getPaginationCount(parent::getDate());
        return view("pages.home", $this->data);
    }
}

```

4.2.5 MemberController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\MembershipRequest;
use App\Models\Membership;
use Illuminate\Http\Request;

class MembershipController extends Controller
{
    private $membershipModel;

    public function __construct()
    {
        $this->membershipModel = new Membership();
    }

    public function becomeMemberRequest(Request $request){
        $memberName = $request->input("memReqName");
        $memberLastname = $request->input("memReqLastname");
        $memberEmail = $request->session()->get("userMembershipInfo")["email"];
        $userId = $request->input("memReqUserId");
        $member = $this->membershipModel->getMember($memberEmail);
        $request->session()->forget("userMembershipInfo");
        if($request->session()->has("successSentEmailMembership")){
            $request->session()->forget("successSentEmailMembership");
        }
        if($request->session()->has("successSentEmailMembership")){
            $request->session()->forget("errorSentEmailMembership");
        }
        if($member){
            return redirect("/membership")->with("error", "Member with same email already exists.");
        }else{
            $code = $this->membershipModel->becomeMember($memberName,
            $memberLastname, $memberEmail, $userId);

```

```

        if($code === 201){
            return redirect("/membership")->with("success", "Your request was
successfully sent.");
        }else{
            return redirect("/membership")->with("error", "There was an error.
Try again later.");
        }
    }
}
}
}
}

```

4.2.6 MenuLinkController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Activity;
use App\Models\MenuModel;
use Illuminate\Http\Request;

class MenuLinkController extends Controller
{
    private $menuModel;
    private $activityModel;

    public function __construct(){
        $this->menuModel = new MenuModel();
        $this->activityModel = new Activity();
    }

    public function deleteLink($id){
        $this->menuModel->deleteLink($id);
        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
deleted comment with id: " . $id . "\t";
        $userId = session()->get('user')->id;
        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        return back();
    }

    public function getLinks(){
        $links = $this->menuModel->getAllLinks();
        return response()->json(["links" => $links], 200);
    }

    public function insertLink(Request $request){
        $text = $request->input("text");
        $path = $request->input("path");
        $parent = $request->input("parent");
        $header = $request->input("header");
        $footer = $request->input("footer");
        $admin = $request->input("admin");
        if($parent == 0){
            $parent = NULL;
        }
        try{
            $id = $this->menuModel->insertLink($text,$path, $parent, $header,
$footer, $admin);

```

```

        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
inserted link with id: ". $id . "\t";
        $userId = session()->get('user')->id;
        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        $links = $this->menuModel->getAllLinks();
        return response()->json(["links" => $links], 201);
    }catch(\PDOException $ex){
        dd($ex->getMessage());
        return response()->json("", 500);
    }
}

public function updateLink(Request $request){
    $id = $request->input("id");
    $text = $request->input("text");
    $path = $request->input("path");
    $parent = $request->input("parent");
    $header = $request->input("header");
    $footer = $request->input("footer");
    $admin = $request->input("admin");
    if($parent == 0){
        $parent = NULL;
    }
    try{
        $this->menuModel->updateLink($id, $text,$path, $parent, $header,
$footer, $admin);
        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
updated link with id: ". $id . "\t";
        $userId = session()->get('user')->id;
        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        return response()->json("", 204);
    }catch(\PDOException $ex){
        dd($ex->getMessage());
        return response()->json("", 500);
    }
}

public function getLinkData($id){
    $link = $this->menuModel->getLinkData($id);
    return response()->json(["link" => $link], 200);
}
}

```

4.2.7 MessageController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Message;
use Illuminate\Http\Request;

class MessageController extends Controller
{
    private $messageModel;

    public function __construct()
    {
        $this->messageModel = new Message();
    }
}

```

```

    }

    public function contactAdmin(Request $request){
        $email = $request->input("emailAddress");
        $question = $request->input("body");
        try{
            $this->messageModel->contactAdmin($email,$question);
            return back()->with("success","Your message was sent.");
        }catch(\PDOException $ex){
            return back()->with("error","There was an error. Try again.");
        }
    }

    public function contactCinema(Request $request){
        $firstname = $request->input("firstnameContCin");
        $lastname = $request->input("lastnameContCin");
        $email = $request->input("emailAddressContCin");
        $question = $request->input("question");
        try{
            $this->messageModel->contactCinema($firstname,$lastname,$email,$question);
            return back()->with("success","Your message was sent.");
        }catch(\PDOException $ex){
            return back()->with("error","There was an error. Try again.");
        }
    }
}

```

4.2.8 MovieController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Movie;
use Illuminate\Http\Request;

class MovieController extends FrontController
{
    private $movieModel;

    public function __construct(){
        parent::__construct();
        $this->movieModel = new Movie();
    }

    public function getSliderData(){
        $sliderData = $this->movieModel->getSliderData();
        return response()->json(['movies' => $sliderData], 200);
    }

    public function getMovies($offset){
        $dataArray = $this->getDate();
        $movies = $this->movieModel->getMovies($dataArray,$offset);
        return response()->json(['movies' => $movies], 200);
    }

    public function filterMovies(Request $request){
        $date = $request->input('date');
    }
}

```

```

        $dateArray = $this->getDate($date);
        $technology = $request->input('technology');
        $sort = $request->input("sort");
        $searched = $request->input("searched");
        $genre = $request->input("genre");
        $movies = $this->movieModel->filterMovies($dateArray, $technology, $sort,
$searched, $genre);
        return response()->json(['movies' => $movies], 200);
    }

    public function getMovie($id){
        $movie = $this->movieModel->getMovie($id);
        $this->data['movie'] = $movie;
        return view("pages.movie_details", $this->data);
    }

    public function getMoviesDdl(Request $request){
        $date = $request->input("date");
        $dateArray = $this->getDate($date);
        $movies = $this->movieModel->getMoviesDdl($dateArray);
        return response()->json(['movies' => $movies], 200);
    }

    public function getDate($date = 0){
        date_default_timezone_set('Europe/Belgrade');
//    $date = 0;
        if($date){
            $todayDate = $date;
            $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
            $todayDate = date('Y-m-d', strtotime($todayDate));
        }else{
            $todayDate = date("Y-m-d", strtotime("2020-12-26"));
            $plus1Day = date('Y-m-d', strtotime($todayDate . ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate . ' - 1 days'));
        }
        $minus1Day = $minus1Day . " 23:59:59";
        $timeNow = " 15:00:00";
        $dateTimeNow = $todayDate . $timeNow;

        $dateArray = [ "plus1Day" => $plus1Day, "minus1Day" => $minus1Day,
"todayDate" => $todayDate, "dateTimeNow" => $dateTimeNow];
        return $dateArray;
    }

    public function rateMovie(Request $request){
        $idMovie = $request->input("idMovie");
        $grade = $request->input("grade");
        $idUser = $request->session()->get('user')->id;
        $userMovieGradeObj = $this->movieModel->getUserMovieGradeObj($idMovie,
$idUser);
//        dd($userMovieGradeObj);
        if($userMovieGradeObj){
            $code = $this->movieModel->rateMovieUpdate($grade, $idMovie, $idUser);
        }else{
            $code = $this->movieModel->rateMovie($grade, $idMovie, $idUser);
        }
        return response()->json("", $code);
    }
}

```

```

    public function comingSoon(){
        $cs = $this->movieModel->comingSoon();
        $this->data['coming_soon'] = $cs;
        return view("pages.coming_soon", $this->data);
    }
}

```

4.2.9 PHPMailerController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\MembershipRequest;
use App\Models\User;
use Illuminate\Http\Request;
use PHPMailer\PHPMailer\PHPMailer;

class PHPMailerController extends FrontController
{
    public function __construct()
    {
        parent::__construct();
    }

    public function becomeMember(MembershipRequest $request){
        $email = $request->input("memEmail");
        $firstname = $request->input("memName");
        $lastname = $request->input("memLastname");
        $userId = $request->input("membershipHidden");
        if(!$userId){
            $userId = 0;
        }
        $userMembershipInfo = ["email" => $email, "firstname" => $firstname,
"lastname" => $lastname, "userId" => $userId];
        $request->session()->put("userMembershipInfo", $userMembershipInfo);

        try{
            $mail = $this->getMailer($email);
            $randomCode = $this->getRandomString();
            $request->session()->put("randomCode", $randomCode);
            $mail->isHTML(true);
// Set email format to HTML
            $mail->Subject = 'MovieBlackout-Become member';
            $mail->Body = 'Your verification code is: '. $randomCode;
// message

            $mail->send();
            $request->session()->put("successSentEmailMembership", "Confirmation
mail was sent to your email address.");
            return view("pages.membership", $this->data);
        }catch (Exception $e) {
            $request->session()->put("errorSentEmailMembership", "There was an
error. Email could not be sent.");
            return back();
        }
    }
}

```



```

public function forgetPassword(Request $request){
    $email = $request->input("forgetPassEmail");

    $request->session()->put("email", $email);
    $userModel = new User();
    $username = $userModel->getUsername($email);
    $request->session()->put("username", $username);
    try{
        $mail = $this->getMailer($email);
        $randomCode = $this->getRandomString();
        $request->session()->put("randomCode", $randomCode);
        $mail->isHTML(true);
// Set email format to HTML
        $mail->Subject = 'MovieBlackout-Password change';
        $mail->Body = 'Your verification code is: '. $randomCode;
// message

        $mail->send();
        $request->session()->put("successForgetPass", "Success.");
        return view("pages.regain_account_access", $this->data);
    }catch (Exception $e) {
        $request->session()->put("errorForgetPass", "Error.");
        return back();
    }
}

}

public function getMailer($email){
    // load Composer's autoloader
    $mail = new PHPMailer(true); // Passing `true`
enables exceptions

    // Server settings
    $mail->SMTPDebug = 0; // Enable
verbose debug output
    $mail->isSMTP(); // Set mailer
to use SMTP
    $mail->Host = 'smtp.gmail.com';
// Specify main and backup SMTP servers
    $mail->SMTPAuth = true; // Enable SMTP
authentication
    $mail->Username = 'valerija.movieblackout@gmail.com'; //
SMTP username
    $mail->Password = 'movie.blackout98'; // SMTP password
    $mail->SMTPSecure = 'tls'; // Enable TLS
encryption, `ssl` also accepted
    $mail->Port = 587; // TCP port to
connect to

    //Recipients
    $mail->setFrom('valerija.movieblackout@gmail.com', 'MovieBlackout');
    $mail->addAddress($email, 'Optional name'); // Add a recipient,
Name is optional
    $mail->addReplyTo('valerija.movieblackout@gmail.com', 'Mailer');
    $mail->addCC($email);
    $mail->addBCC($email);
    return $mail;
}

```

```

    public function getRandomString(){
        $pool = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
        $randomCode = substr(str_shuffle(str_repeat($pool, 5)), 0, 10);
        return $randomCode;
    }
}

```

4.2.10 ProjectionController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Projection;
use Illuminate\Http\Request;

class ProjectionController extends Controller
{
    private $projectionModel;

    public function __construct(){
        $this->projectionModel = new Projection();
    }

    public function getProjectionsDdl(Request $request){
        $date = $request->input("date");
        $dateArray = $this->getDate($date);//$date !!!
        $movieId = $request->input("movieId");
        $projections = $this->projectionModel->getProjectionsDdl($dateArray,
        $movieId);
        return response()->json(['projections' => $projections], 200);
    }

    public function getDate($date = 0)
    {
        date_default_timezone_set('Europe/Belgrade');
        // $date = 0;
        if($date){
            $todayDate = $date;
            $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
            $todayDate = date('Y-m-d', strtotime($todayDate));
        }else{
            $todayDate = date("Y-m-d", strtotime("2020-12-26"));
            $plus1Day = date('Y-m-d', strtotime($todayDate . ' + 1 days'));
            $minus1Day = date('Y-m-d', strtotime($todayDate . ' - 1 days'));
        }
        $minus1Day = $minus1Day . " 23:59:59";
        $timeNow = " 15:00:00";
        $dateTimeNow = $todayDate . $timeNow;

        $dateArray = [ "plus1Day" => $plus1Day, "minus1Day" => $minus1Day,
        "todayDate" => $todayDate, "dateTimeNow" => $dateTimeNow];
        return $dateArray;
    }
}

```

```
}  
}
```

4.2.11 ReservationController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Projection;  
use App\Models\Reservation;  
use Illuminate\Http\Request;  
  
class ReservationController extends FrontController  
{  
    private $reservationModel;  
    private $projectionModel;  
  
    public function __construct(){  
        parent::__construct();  
        $this->reservationModel = new Reservation();  
        $this->projectionModel = new Projection();  
    }  
  
    public function continueReservation(Request $request){  
        $movieId = $request->input("reservationMovieDdl");  
        $projectionId = $request->input("reservationProjectionDdl");  
        $date = $request->input("reservationDateDdl");  
        $dayOfWeek = $this->getWeekday($date);  
        $dayOfWeek = intval($dayOfWeek);  
        // dd($dayOfWeek);  
        $seats = $request->input("numSeats");  
        $projectionInfo = $this->projectionModel->  
>getProjection($projectionId);//$movieId,  
        // dd($projectionInfo);  
        $price = $this->reservationModel->calculetePriceReservation($dayOfWeek,  
$projectionInfo->idTechnology)->price;  
        $pricePerTicket = $price;  
        $price *= $seats;  
        $this->data["pricePerTicket"] = $pricePerTicket;  
        $this->data["projectionInfo"] = $projectionInfo;  
        $this->data["price"] = $price;  
        $this->data["seats"] = $seats;  
        $this->data['hallRows'] = $this->reservationModel->  
>getHallRows($projectionInfo->idHall,$projectionId);  
        return view("pages.reservation", $this->data);  
    }  
  
    public function getWeekday($date = 0){  
        date_default_timezone_set('Europe/Belgrade');  
        if($date){  
            $weekday = date('w', strtotime($date));  
            if($weekday === '0'){//Sunday  
                $weekday = '7';  
            }  
        }  
  
        return $weekday;// - 1  
    }  
}
```

```

    public function makeReservation(Request $request){
        $seatsNum = $request->input("seatsNum");
        $seatIdArray = $request->input("seatIdArray");
        $projectionId = $request->input("projectionId");
        $userId = $request->session()->get('user')->id;
        date_default_timezone_set('Europe/Belgrade');
        $todayDate = date("Y-m-d H:i:s");
        $code = $this->reservationModel->makeReservation($seatsNum,
        $projectionId,$userId, $todayDate, $seatIdArray);
        return response()->json("", $code);
    }

    public function continueProjectionReservation($projectionId){
        $projectionInfo = $this->projectionModel->getProjection($projectionId);
        $dayOfWeek = $this->getWeekday($projectionInfo->starts_at);
        $dayOfWeek = intval($dayOfWeek);
        $pricePerTicket = $this->reservationModel-
>calculatePriceReservation($dayOfWeek, $projectionInfo->idTechnology)->price;
        $price = $pricePerTicket * 1;//default seats num is 1
        $this->data["pricePerTicket"] = $pricePerTicket;
        $this->data["projectionInfo"] = $projectionInfo;
        $this->data["price"] = $price;
        $this->data["seats"] = 1;
        $this->data['hallRows'] = $this->reservationModel-
>getHallRows($projectionInfo->idHall,$projectionId);
        return view("pages.reservation", $this->data);
    }
}

```

4.2.12 UserController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Http\Requests\RegistrationRequest;
use App\Models\Activity;
use App\Models\User;
use Illuminate\Http\Request;

class UserController extends FrontController
{
    private $userModel;
    private $activityModel;

    public function __construct()
    {
        parent::__construct();
        $this->userModel = new User();
        $this->activityModel = new Activity();
    }

    public function deleteUser($id){
        $this->userModel->deleteUser($id);
        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
deleted user with id: ". $id . "\t";
        $userId = session()->get('user')->id;
    }
}

```

```

        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        return back();
    }

    public function getUsers(){
        $users = $this->userModel->getUsers();
        return response()->json(["users" => $users], 200);
    }

    public function updateUser(Request $request){
        // dd($request);
        $id = $request->input("id");
        $username = $request->input("username");
        $password = $request->input("password");
        $email = $request->input("email");
        $gender = $request->input("gender");
        $role = $request->input("role");
        $news = $request->input("news");
        try{
            $this->userModel->updateUser($id, $username, $password, $email,
            $gender, $role, $news);
            $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
            updated user with id: ". $id . "\t";
            $userId = session()->get('user')->id;
            $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
            return response()->json("", 204);
        }catch(\PDOException $ex){
            // dd($ex->getMessage());
            return response()->json("", 500);
        }
    }

    public function getUser($id){
        $user = $this->userModel->getUser($id);
        return response()->json(["user" => $user], 200);
    }

    public function accountAccess(Request $request){
        $code = $request->session()->get("randomCode");
        $userCode = $request->input("accountAccessCode");
        $newPass = $request->input("accountAccessPass");
        $email = $request->session()->get('email');

        if($code === $userCode){
            try{
                $this->userModel->changePassword($email, $newPass);
                $this->data['successfullyChangedPass'] = "Your password was
                changed successfully.";
                return view("pages.regain_account_access", $this->data);
            }catch(\PDOException $ex){
                return redirect()->back()->with("error", "There was an error.");
            }
        }
    }

    public function login(LoginRequest $request){
        $username = $request->input("logUsername");
        $password = $request->input("logPass");
    }

```

```

        $user = $this->userModel->getUserByUsernamePassword($username, $password);

        if($user){
            $request->session()->put("user", $user);
            if($user->role_id === 1){
                return \redirect("/home");
            }else if($user->role_id === 2){
                return \redirect("/admin");
            }
        }else{
            return redirect("/home")->with("message", "You are not registered.");
        }
    }

    public function logout(Request $request){
        $request->session()->forget("user");
        return redirect("/home");
    }

    public function registration(RegistrationRequest $request){
        $username = $request->input("username");
        $password = $request->input("pass");
        $tel = $request->input("tel");
        $email = $request->input("email");
        $selectedGender = $request->input("selectedGender");
        $sendViaMail = $request->input("sendViaMail");
        // dd($request);
        $code = $this->userModel->registerUser($username, $password, $tel, $email,
        $selectedGender, $sendViaMail);

        return response()->json('', $code);
    }
}

```

Admin

4.2.13 ActivityController.php

```

<?php

namespace App\Http\Controllers\admin;

use App\Http\Controllers\Controller;
use App\Models\Activity;
use Illuminate\Http\Request;

class ActivityController extends Controller
{
    public function sortActivity(Request $request){
        $am = new Activity();
        $act = $am->sortActivity($request->input("sort"));
        return response()->json(["activities" => $act], 200);
    }
}

```

4.2.14 MovieController.php

```
<?php

namespace App\Http\Controllers\admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\InsertMovieRequest;
use App\Http\Requests\UpdateMovieRequest;
use App\Models\Activity;
use App\Models\Movie;
use App\Models\Technology;
use Illuminate\Http\Request;

class MovieController extends Controller
{
    private $movieModel;
    private $technologyModel;
    private $activityModel;

    public function __construct()
    {
        $this->movieModel = new Movie();
        $this->technologyModel = new Technology();
        $this->activityModel = new Activity();
    }

    // public function pokusaj(Request $request){
    //     $response = cloudinary()->upload($request->file('pokusaj')->getRealPath()->getSecurePath());
    //     dd($response);
    // }

    public function deleteMovie($id){
        try{
            $this->movieModel->deleteMovie($id);
            $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
deleted movie with id: " . $id . "\t";
            $userId = session()->get('user')->id;
            $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
            return back();
        }catch (\PDOException $ex){
            return back()->with("errorDel","There was an error.");
        }
    }

    public function getMovieTechnologies($movieId){
        $technologies = $this->technologyModel->getMovieTechnologies($movieId);
        return response()->json(["technologies" => $technologies], 200);
    }

    public function insertMovie(InsertMovieRequest $request){
        //InsertMovieRequest
        // dd($request);
        $name = $request->input("nameMovIns");
        $running_time = $request->input("timeMovIns");
        $desc = $request->input("taShortDescIns");
        $detailed_desc = $request->input("taDetailedDescIns");
        $distributer = $request->input("distributerMovIns");
    }
}
```

```

$director = $request->input("directorMovIns");
$technology = $request->input("technologiesMovIns");
$technologyArray = [];
if($technology == 3){
    $techns = $this->technologyModel->getTechnologies();
    foreach($techns as $t){
        $technologyArray[] = $t->id;
    }
}

}else{
    $technologyArray[] = $technology;
}
$country = $request->input("countryMovIns");
$date = $request->input("dateMovIns");
$actors = array_unique($request->input("actorInsertHidden"));
$genres = array_unique($request->input("genreInsertHidden"));
$age_limit = $request->input("ageMovIns");
$cs = $request->input("csMovIns");
$activeSlider = $request->input("activeSliderMovIns");

$picIds = [];
$imgName = "picMovIns";
// dd($request->file("picMovIns")->getRealPath());
$picId = 0;
if (!empty($_FILES[$imgName]['name'])) {
    $picId = $this->processImg($request->file("picMovIns")->getRealPath(),$imgName);
    if($picId){
        $picIds["picture_id"] = $picId;
    }
}
$imgNameBg = "picMovBgIns";
$picBgId = 0;
if (!empty($_FILES[$imgNameBg]['name'])) {
    $picBgId = $this->processImg($request->file("picMovBgIns")->getRealPath(),$imgNameBg);
    if($picBgId){
        $picIds["about_movie_pic_id"] = $picBgId;
    }
}
$imgNameSlider = "picSliderMovUpd";
$picSliderId = 0;
if (!empty($_FILES[$imgNameSlider]['name'])) {
    $picSliderId = $this->processImg($request->file("picSliderMovIns")->getRealPath(),$imgNameSlider); //ova funkcija ne obradjuje sliku
    if($picSliderId){
        $picIds["slider_picture_id"] = $picSliderId;
    }
}

try{
    $movieId = $this->movieModel->insertMovie($name, $running_time, $desc,
    $detailed_desc, $distributer, $director,$country, $date, $actors, $genres,
    $age_limit, $cs);
    $this->movieModel->insertMovieImages($movieId, $picIds, $activeSlider);
    // dd($technologyArray);
    foreach($technologyArray as $t){
        $this->technologyModel->insertTechnologyMovie($t, $movieId);
    }
}

```



```

        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
inserted movie with id: ". $movieId . "\t";
        $userId = session()->get('user')->id;
        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        return back()->with("successIns", "New movie was inserted.");
    }catch(\PDOException $ex){
//        dd($ex->getMessage());
        return back()->with("errorIns", "There was an error.");
    }
}

public function updateMovie(UpdateMovieRequest $request){
//    dd($request);
    $id = $request->input("movieIdHidden");
    $name = $request->input("nameMovUpd");
    $running_time = $request->input("timeMovUpd");
    $desc = $request->input("taShortDesc");
    $detailed_desc = $request->input("taDetailedDesc");
    $distributer = $request->input("distributerMovUpd");
    $director = $request->input("directorMovUpd");
    $technology = $request->input("technologiesMovUpd");
    $technologyArray = [];
    if($technology == 3){
        $techns = $this->technologyModel->getTechnologies();
        foreach($techns as $t){
            $technologyArray[] = $t->id;
        }
    }else{
        $technologyArray[] = $technology;
    }
    $country = $request->input("countryMovUpd");
    $date = $request->input("dateMovUpd");
    $actors = array_unique($request->input("actorHidden"));
    $genres = array_unique($request->input("genreHidden"));
    $age_limit = $request->input("ageMovUpd");
    $cs = $request->input("csMovUpd");
    $activeSlider = $request->input("activeSliderMovUpd");

    $picIds = [];
    $imgName = "picMovUpd";
    $picId = 0;
    if (!empty($_FILES[$imgName]['name'])) {
        $picId = $this->processImg($request->file("picMovUpd")-
>getRealPath(), $imgName);
        if($picId){
            $picIds["picture_id"] = $picId;
        }
    }
    $imgNameBg = "picMovBgUpd";
    $picBgId = 0;
    if (!empty($_FILES[$imgNameBg]['name'])) {
        $picBgId = $this->processImg($request->file("picMovBgUpd")-
>getRealPath(), $imgNameBg);
        if($picBgId){
            $picIds["about_movie_pic_id"] = $picBgId;
        }
    }
    $imgNameSlider = "picSliderMovUpd";
    $picSliderId = 0;

```

```

        if (!empty($_FILES[$imgNameSlider]['name'])) {
            $picSliderId = $this->processImg($request->file("picSliderMovUpd")-
>getRealPath(),$imgNameSlider);//ova funkcija ne obradjuje sliku
            if($picSliderId){
                $picIds["slider_picture_id"] = $picSliderId;
            }
        }
        try{
            $this->movieModel->updateMovie($id, $name, $running_time, $desc,
            $detailed_desc, $distributer, $director,$country, $date, $actors, $genres,
            $age_limit, $cs);
            $this->movieModel->updateMovieImages($id, $picIds, $activeSlider);
            $this->technologyModel->deleteTechnologyMovie($id);
            foreach($technologyArray as $t){
                $this->technologyModel->insertTechnologyMovie($t, $id);
            }
            $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
updated movie with id: ". $id . "\t";
            $userId = session()->get('user')->id;
            $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
            return back()->with("success","Changes were made successfully.");
        }catch(\PDOException $ex){
            return back()->with("error","There was an error");
        }
    }

}

public function getMovieData(Request $request){
    $id = $request->input("id");
    $movie = $this->movieModel->getMovieData($id);
    return response()->json(["movie" => $movie], 200);
}

public function getMovies($offset, $searched){
    $movies = $this->movieModel->getMoviesAdmin($offset, $searched);
    //dd($movies);
    $allMovies = $this->movieModel->getAllMoviesAdmin();
    return response()->json(["movies"=>$movies,"allMovies" => $allMovies],
200);
}

public function processImg($imgPath, $fileName){
    $picName = $_FILES[$fileName]['name'];
    $extension = strtolower(explode(".", $picName)[1]);
    $type = "";
    if($extension == 'jpg'){
        $type = "image/jpeg";
    }else if($extension == 'png'){
        $type = "image/png";
    }else if($extension == 'jpeg'){
        $type = "image/jpeg";
    }
    $finalFileName = time() . "_" . $picName;
    $alt = explode(".", $picName);
    $alt = $alt[0];

    $smallerFileName = 'edited_' . $finalFileName;
    //    $path = $folder . $finalFileName;

```

```

        try{
            $path = cloudinary()->upload( $_FILES[$fileName]['tmp_name'])-
>getSecurePath();
            // dd($path);
            $imgArray = explode("/", $path);
            $cloudPicName = end($imgArray);
            array_pop($imgArray);
            $imgPathString = implode("/", $imgArray);
            $imgPathString .= "/";
            $picId = $this->movieModel->insertImage($cloudPicName, $alt,
$type, $imgPathString);
            return $picId;
        }catch (\PDOException $ex){
            return back()->with("error", "There was an error");
        }
    }

    }catch (\PDOException $ex){
        return back()->with("error", "There was an error");
    }
}
}
}
}

```

4.2.15 ProjectionController.php

```

<?php

namespace App\Http\Controllers\admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\InsertProjectionsRequest;
use App\Http\Requests\UpdateProjectionRequest;
use App\Models\Activity;
use App\Models\Projection;
use App\Models\Technology;
use Illuminate\Http\Request;

class ProjectionController extends Controller
{
    private $projectionModel;
    private $technologyModel;
    private $activityModel;

    public function __construct()
    {
        $this->projectionModel = new Projection();
        $this->technologyModel = new Technology();
        $this->activityModel = new Activity();
    }

    public function deleteProjection($id){
        $this->projectionModel->deleteProjection($id);
        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username . "
deleted projection with id: ". $id . "\t";
        $userId = session()->get('user')->id;
        $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
        return back();
    }
}

```

```

    }

    public function updateProjection(UpdateProjectionRequest $request){
        $idProjection = $request->input("idProjHidden");
        // dd($idProjection);
        $movie = $request->input("movieProjectionUpdate");
        $date = $request->input("dateProjectionUpdate");
        $time = $request->input("timeProjectionUpdate");
        $technology = $request->input("technProjectionUpdate");
        $reservation = $request->input("resProjectionUpdate");
        $hall = $request->input("hallProjectionUpdate");
        $dateTimeProj = $date . " " . $time . ":00";
        $dataArray = $this->getDate($dateTimeProj);

        $projections = $this->projectionModel->projectionAvailable($hall,
        $dataArray, $idProjection);
        // dd($projections);
        $available = 1;
        foreach($projections as $p){
            if($p->starts_at == $dateTimeProj){
                $available = 0;
                break;
            }else if($p->starts_at > $dateTimeProj) {
                $whenWouldOurProjEnd = $this->whenWouldOurProjEnd($dateTimeProj,
        $p->running_time);
        // dd($whenWouldOurProjEnd);
                if ($p->starts_at <= $whenWouldOurProjEnd) {
                    $available = 0;
                    break;
                }
            }else if($p->hallFreeAt >= $dateTimeProj){//za projekcije koje pocinju
        pre izbranog vremena za novu projekciju
        // dd("ggg");
                $available = 0;
                break;
            }
        }
        if($available){
            try{
                $id = $this->projectionModel->updateProjection($idProjection,
        $movie, $dateTimeProj, $hall, $technology, $reservation);
                $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')->username
        . " updated projection with id: " . $idProjection . "\t";
                $userId = session()->get('user')->id;
                $this->activityModel->write($text, date("Y.m.d H:i:s"), $userId);
                return back()->with("successUpdProj", "Projection was updated
        successfully.");
            }catch(\PDOException $ex){
        // dd($ex->getMessage());
                return back()->with("errorUpdProj", "There was an error.");
            }
        }else{
            return back()->with("errorUpdProj", "Chosen hall is not available.");
        }
    }

    public function whenWouldOurProjEnd($date, $running_time){
        $timeToAdd = $running_time + 30;
        $addedMinutes = strtotime("+ " . $timeToAdd . " minutes", strtotime($date));
    }

```

```

        return date('Y-m-d H:i:s', $addedMinutes);
    }

    public function getProjections(Request $request){
        $searched = $request->input("searched");
        $projections = $this->projectionModel->getProjections($searched);
        return response(["projections" => $projections], 200);
    }

    public function getProjectionData($id){
        $projection = $this->projectionModel->getProjection($id);
        $movieTechn = $this->technologyModel->getMovieTechnologiesProjection($id);
        return response(["projection" => $projection, "technologies" =>
$movieTechn], 200);
    }

    public function insertProjections(InsertProjectionsRequest $request){
//
        dd($request);
        $projectionMovies = $request->input("projectionMovie");
        $projectionDates = $request->input("projectionDate");
        $projectionTimes = $request->input("projectionTime");
        $projectionTechnologies = $request->input("projectionTechnology");
        $projectionHalls = $request->input("projectionHall");
        $projectionsResAvailable = $request->input("projectionResAvailable");
        $dateTimeProjs = [];
        for($i = 0; $i < count($projectionMovies); $i++){
            $dateTimeProj = $projectionDates[$i] . " " . $projectionTimes[$i] .
":00";
            $dateTimeProjs[] = $dateTimeProj;
            $dateArray = $this->getDate($dateTimeProj);
            $projections = $this->projectionModel-
>projectionAvailable($projectionHalls[$i], $dateArray);
            $available = 1;
            foreach($projections as $p){
                if($p->starts_at == $dateTimeProj){
                    $available = 0;
                    break;
                }else if($p->starts_at > $dateTimeProj) {
                    $whenWouldOurProjEnd = $this-
>whenWouldOurProjEnd($dateTimeProj, $p->running_time);
//
                    dd($whenWouldOurProjEnd);
                    if ($p->starts_at <= $whenWouldOurProjEnd) {
                        $available = 0;
                        break;
                    }
                }else if($p->hallFreeAt >= $dateTimeProj){//za projekcije koje
pocinju pre izbranog vremena za novu projekciju
                    $available = 0;
                    break;
                }
            }
            if(!$available) {
                break;
            }
        }
        if($available){
            try{
                $userId = session()->get('user')->id;
                for($i = 0; $i < count($projectionMovies); $i++) {

```

```

        $id = $this->projectionModel->
>insertProjections($projectionMovies[$i], $dateTimeProjs[$i],
$projectionHalls[$i], $projectionTechnologies[$i], $projectionsResAvailable[$i]);
        $text = $_SERVER['REMOTE_ADDR'] . "\t" . session('user')-
>username . " inserted projection with id: " . $id . "\t";
        $this->activityModel->write($text, date("Y.m.d H:i:s"),
$userId);
    }
    return back()->with("successProjectionInsert", "Projections were
inserted successfully.");
} catch (\PDOException $ex) {
    dd($ex->getMessage());
    return back()->with("errorProjectionInsert", "There was an
error.");
}
} else {
    return back()->with("errorProjectionInsert", "Chosen hall is not
available.");
}
}

public function getDate($date = 0){
    date_default_timezone_set('Europe/Belgrade');
//    $date = 0; //obrisati
    if($date){
        $todayDate = $date;
        $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
        $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
        $todayDate = date('Y-m-d', strtotime($todayDate));
    } else {
        $todayDate = date("Y-m-d", strtotime("2020-12-26"));
        $plus1Day = date('Y-m-d', strtotime($todayDate. ' + 1 days'));
        $minus1Day = date('Y-m-d', strtotime($todayDate. ' - 1 days'));
    }
    $minus1Day = $minus1Day . " 23:59:59";
    $timeNow = " 15:00:00";
    $dateTimeNow = $todayDate . $timeNow;

    $dateArray = [ "plus1Day" => $plus1Day, "minus1Day" => $minus1Day,
"todayDate" => $todayDate, "dateTimeNow" => $dateTimeNow];
    return $dateArray;
}
}

```

4.3 Models

4.3.1 AboutCinema.php

```

<?php

namespace App\Models;

class AboutCinema
{
    public function getAboutCinemaText(){
        return \DB::table("about_cinema")

```

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

4.3.2 Activity.php

```
<?php

namespace App\Models;

class Activity
{
    public function write($write, $date, $userId){
        return \DB::table("activity")
            ->insertGetId([
                "id" => NULL,
                "text" => $write,
                "activity_time" => $date,
                "user_id" => $userId
            ]);
    }

    public function print(){
        return \DB::table("activity")
            ->orderBy("activity_time", "DESC")
            ->get();
    }

    public function sortActivity($sort){
        return \DB::table("activity")
            ->orderBy("activity_time", $sort)
            ->get();
    }
}
```

4.3.3 Actor.php

```
<?php

namespace App\Models;

class Actor
{
    public function getActors(){
        return \DB::table("actor")
            ->select("name", "id")
            ->get();
    }
}
```

4.3.4 Author.php

```
<?php

namespace App\Models;

class Author
{
    public function getAuthorInfo(){
        return \DB::table("author")
            ->first();
    }
}
```

4.3.5 Comment.php

```
<?php

namespace App\Models;

class Comment
{
    private $limitComments = 5;
    // private static $offsetComments = 5;

    public function getAllComments(){
        return \DB::table("comment AS c")
            ->join("user AS u", "c.user_id", "u.id")
            ->join("movie AS m", "c.movie_id", "m.id")
            ->select("c.id", "c.text", "c.inserted_at AS
posted_at", "c.updated_at", "u.username", "u.gender_id", "u.registered_at", "c.user_id"
, "m.id AS idMovie")
            ->where([
                ["c.is_deleted", 0]
            ])
            ->orderBy("c.inserted_at", "desc")
            ->get();
    }

    public function viewMoreComments($id, $offset){
        $com = \DB::table("comment AS c")
            ->join("user AS u", "c.user_id", "u.id")
            ->join("movie AS m", "c.movie_id", "m.id")
            ->select("c.id", "c.text", "c.inserted_at AS
posted_at", "c.updated_at", "u.username", "u.gender_id", "u.registered_at", "c.user_id"
, "m.id AS idMovie")
            ->where([
                ["m.id", $id],
                ["c.is_deleted", 0],
                ["c.parent_id", NULL]
            ])
            ->offset($offset)
            ->limit($this->limitComments)
            ->orderBy("c.inserted_at", "desc")
            ->get();
    }
}
```



```

        return $com;
    }

    public function deleteComment($id){
        return \DB::table("comment")
            ->where("id",$id)
            ->update([
                "is_deleted" => 1
            ]);
    }

    public function updateComment($modifiedCommText, $commentId, $dateTime){
        $data = [];
        $code = 200;
        try{
            $success = \DB::table("comment")
                ->where("id", $commentId)
                ->update([
                    "text" => $modifiedCommText,
                    "updated_at" => $dateTime
                ]);

            if($success){
                $code = 204;
            }
        }catch (\PDOException $ex){
            $code= 409;
            $ex->getMessage() ;
        }
        $data["code"] = $code;
        return $data;
    }

    public function getRepliesOfComment($parent_id){
        $com = \DB::table("comment AS c")
            ->join("user AS u", "c.user_id", "u.id")
            ->join("movie AS m", "c.movie_id", "m.id")
            ->select("c.id", "c.text", "c.inserted_at AS
posted_at", "u.username", "u.gender_id", "u.registered_at", "c.user_id", "c.movie_id")
            ->where([
                ["c.parent_id", $parent_id],
                ["c.is_deleted", 0],
            ])
            ->orderBy("c.inserted_at", "desc")
            ->get();
        return $com;
    }

    public function postComment($comment, $idMovie, $idUser, $dateTime,
$parent_id = NULL)
    {
        $code = 200;
        $data = [];
        try {
            $idInserted = \DB::table("comment")
                ->insertGetId([
                    "id" => NULL,

```

```

        "text" => $comment,
        "parent_id" => $parent_id,
        "user_id" => $idUser,
        "movie_id" => $idMovie,
        "inserted_at" => $dateTime,
        "updated_at" => NULL,
        "is_deleted" => 0
    ]);
    if($idInserted){
        $code = 201;
        $newComment = $this->getNewlyPostedComment($idInserted);
        $data["newComment"] = $newComment;
    }else{
        $code= 409;
    }
} catch (\PDOException $ex){
    $code= 409;
    $ex->getMessage();
}
$data["code"] = $code;
return $data;
}

public function getNewlyPostedComment($idInserted){
    return \DB::table("comment AS c")
        ->join("user AS u", "c.user_id", "u.id")
        ->join("movie AS m", "c.movie_id", "m.id")
        ->select("c.id", "c.text", "c.inserted_at AS
posted_at", "u.username", "u.gender_id", "u.registered_at", "c.user_id")
        ->where("c.id", $idInserted)
        ->first();
}
}

```

4.3.6 Country.php

```

<?php

namespace App\Models;

class Country
{
    public function getCountries(){
        return \DB::table("country")
            ->select("name", "id")
            ->get();
    }
}

```

4.3.7 Director.php

```

<?php

namespace App\Models;

```

```

class Director
{
    public function getDirectors(){
        return \DB::table("director")
            ->select("name","id")
            ->get();
    }
}

```

4.3.8 Distributer.php

```

<?php

namespace App\Models;

class Distributer
{
    public function getDistributers(){
        return \DB::table("distributer")
            ->select("name","id")
            ->get();
    }
}

```

4.3.9 Gender.php

```

<?php

namespace App\Models;

class Gender
{
    public function getGenders(){
        return \DB::table("user_genre")
            ->get();
    }
}

```

4.3.10 Genre.php

```

<?php

namespace App\Models;

class Genre
{
    public function getGenres(){
        return \DB::table("genre")
            ->get();
    }
}

```

```
}  
}
```

4.3.11 Hall.php

```
<?php  
  
namespace App\Models;  
  
class Hall  
{  
    public function getHalls(){  
        return \DB::table("hall")  
            ->get();  
    }  
}
```

4.3.12 Membership.php

```
<?php  
  
namespace App\Models;  
  
use http\Client\Request;  
  
class Membership  
{  
    public function becomeMember($memberName, $memberLastname, $memberEmail,  
$userId){  
        $code = 200;  
        $query = [  
            "id" => NULL,  
            "firstname" => $memberName,  
            "lastname" => $memberLastname,  
            "email" => $memberEmail,  
            "user_id" => $userId  
        ];  
        if(!$userId){  
            $query["user_id"] = NULL;  
        }  
        try{  
            $id = \DB::table("member")  
                ->insertGetId($query);  
            if($id){  
                $code = 201;  
            }else{  
                $code = 409;  
            }  
        }catch (\PDOException $ex){  
            $ex->getMessage();  
            $code = 409;  
        }  
        return $code;  
    }  
}
```

```

    }

    public function getMember($memberEmail){
        return \DB::table("member")
            ->where("email", $memberEmail)
            ->first();
    }
}

```

4.3.13 MenuModel.php

```

<?php

namespace App\Models;

class MenuModel
{
    public function getMenuLinks(){
        $links = \DB::table("menu_links AS ml")
            ->leftJoin("footer_headline AS fh","footer_headline_id","fh.id")
            ->select("ml.id AS menu_id", "ml.text","ml.href", "ml.parent_id",
                "in_footer", "in_header", "ml.footer_headline_id", "fh.*")
            ->where([
                ["ml.admin_link",0],
                ["ml.is_deleted",0]
            ])
            ->get();

        foreach($links as $link) {
            $link->childrenLinks = \DB::table("menu_links AS ml")
                ->leftJoin("footer_headline AS
                fh","footer_headline_id","fh.id")
                ->where("parent_id","=", $link->menu_id)
                ->select("ml.id AS menu_id",
                "ml.text","ml.href", "ml.parent_id", "in_footer", "in_header",
                "ml.footer_headline_id", "fh.*")
                ->get();
        }
        return $links;
    }

    //ADMIN

    public function deleteLink($id){
        return \DB::table("menu_links")
            ->where("id", $id)
            ->update([
                "is_deleted" => 1
            ]);
    }

    public function insertLink($text,$path, $parent, $header, $footer, $admin){
        return \DB::table("menu_links")
            ->insertGetId([
                "id" => NULL,
                "text" => $text,
                "href" => $path,

```

```

        "in_header" => $header,
        "in_footer" => $footer,
        "parent_id" => $parent,
        "admin_link" => $admin,
        "inserted_at" => date("Y-m-d H:i:s"),
        "updated_at" => NULL,
        "is_deleted" => 0
    ]);
}

public function updateLink($id, $text, $path, $parent, $header, $footer,
$admin){
    return \DB::table("menu_links")
        ->where("id", $id)
        ->update([
            "text" => $text,
            "href" => $path,
            "in_header" => $header,
            "in_footer" => $footer,
            "parent_id" => $parent,
            "admin_link" => $admin
        ]);
}

public function getMenuLinksAdmin(){
    return \DB::table("menu_links AS ml")
        ->select("ml.id AS menu_id", "ml.text", "ml.href")
        ->where([
            ["ml.admin_link", 1],
            ["ml.is_deleted", 0]
        ])
        ->get();
}

public function getAllLinks(){
    return \DB::table("menu_links AS ml")
        ->select("ml.id AS menu_id",
"ml.text", "ml.href", "ml.in_footer", "ml.in_header", "ml.admin_link")
        ->where([
            ["ml.is_deleted", 0]
        ])
        ->get();
}

public function getLinkData($id){
    return \DB::table("menu_links AS ml")
        ->leftJoin("menu_links AS ml1", "ml.parent_id", "ml1.id")
        ->select("ml.id AS menu_id",
"ml.text", "ml.href", "ml.in_footer", "ml.in_header", "ml.admin_link", "ml1.id AS
idParent", "ml1.text AS textParent")
        ->where([
            ["ml.id", $id]
        ])
        ->get();
}
}

```

4.3.14 Message.php

```
<?php

namespace App\Models;

class Message
{
    public function contactAdmin($email,$question){
        return \DB::table("message")
            ->insertGetId([
                "id" => NULL,
                "email" => $email,
                "text" => $question,
                "firstname" => NULL,
                "lastname" => NULL,
                "is_read" => 0
            ]);
    }

    public function contactCinema($firstname,$lastname,$email,$question){
        return \DB::table("message")
            ->insertGetId([
                "id" => NULL,
                "email" => $email,
                "text" => $question,
                "firstname" => $firstname,
                "lastname" => $lastname,
                "is_read" => 0
            ]);
    }
}
```

4.3.15 Movie.php

```
<?php

namespace App\Models;

use DateInterval;
use DateTime;

class Movie
{
    private $limit = 4;
    private $limitComments = 5;
    private $adminLimit = 5;

    public function rateMovie($grade, $idMovie, $idUser){
        $code= 200;
        try{
            $id = \DB::table("user_movie_grade")
                ->insertGetId([
                    "id" => NULL,
                    "user_id" => $idUser,
```

```

        "movie_id" => $idMovie,
        "grade" => $grade
    });
    if($id){
        $code= 201;
    }else{
        $code = 409;
    }
}catch(\PDOException $ex){
    $ex->getMessage();
    $code = 409;
}
return $code;
}

public function rateMovieUpdate($grade, $idMovie, $idUser){
    $code = 200;
    try{
        \DB::table("user_movie_grade")
            ->where([
                ["movie_id", $idMovie],
                ["user_id", $idUser]
            ])
            ->update([
                "grade" => $grade
            ]);
        $code = 204;
    }catch(\PDOException $ex){
        $ex->getMessage();
        $code = 409;
    }
    return $code;
}

public function getUserMovieGradeObj($idMovie, $idUser){
    return \DB::table("user_movie_grade as umg")
        ->where([
            ["movie_id", $idMovie],
            ["user_id", $idUser]
        ])
        ->first();
}

public function filterMovies($dateArray, $technology, $sort, $searched,
$genre){
    $continueQuery = $this->makeContinueQuery($dateArray["plus1Day"],
$dateArray["minus1Day"],$technology, $searched, $genre);

    $movies = \DB::table("movie AS m")
        ->join("movie_picture AS mp","m.id","mp.movie_id")
        ->join("picture AS p","mp.picture_id","p.id")
        ->join("projection as pro","pro.movie_id","m.id")
        ->join("movie_technology AS mt","pro.movie_technology_id","mt.id")
        ->join("movie_genre AS mg", "mg.movie_id", "m.id")
        ->join("genre AS g", "mg.genre_id", "g.id")
        ->select("m.id AS idMovie","m.description","m.name AS
movieName","m.running_time","m.age_limit","m.description","m.in_cinemas_from","p.p
ath","p.name AS picName","p.alt")
        ->where($continueQuery);
}

```



```

        $movies = $movies-
>groupBy("idMovie","m.description","movieName","m.running_time","m.age_limit","m.d
escription","m.in_cinemas_from","p.path","picName","p.alt" )
        ->orderBy("m.name", $sort)
        ->get();

        foreach($movies as $m){
            $m->grade = $this->getGrade($m->idMovie);
            $m->projections = $this->getProjections($m->idMovie, $dateArray,
intval($technology));
            $m->genres = $this->getGenres($m->idMovie);
        }
        // dd($movies);
        return $movies;
    }

    public function makeContinueQuery($plus1Day, $minus1Day,$technology,
$searched, $genre){
        // dd($minus1Day);
        $whereArray = [];
        array_push($whereArray, ["pro.starts_at","<",$plus1Day],
["pro.starts_at",">",$minus1Day],

["m.is_deleted",0],["m.in_cinema",1],["pro.is_deleted",0]);

        if($technology != '0'){
            array_push($whereArray, ["mt.technology_id", intval($technology)]);
        }
        if($genre != '0'){
            array_push($whereArray, ["g.id", intval($genre)]);
        }
        if($searched != ''){
            $stringToSearch = "%$searched%";
            array_push($whereArray, ["m.name", "LIKE" ,$stringToSearch]);
        }
        return $whereArray;
    }

    public function getSliderData(){
        $movies = \DB::table("movie AS m")
            ->join("movie_picture AS mp","m.id","mp.movie_id")
            ->join("picture AS p","mp.slider_picture_id","p.id")
            ->join("director AS d","m.director_id","d.id")
            ->join("user_movie_grade AS umg","umg.movie_id","m.id")
            ->select("m.id AS idMovie","m.name AS
movieName","m.running_time","m.in_cinemas_from","p.path","p.name AS
picName","p.alt", "d.name AS dirName", \DB::raw("COUNT(*) AS voters"))
            ->where("mp.is_active_slider", 1)
            ->groupBy("idMovie",
"movieName","m.running_time","m.in_cinemas_from","p.path", "picName","p.alt",
"dirName")
            ->get();
        foreach($movies as $m){
            $m->grade = $this->getGrade($m->idMovie);
            $m->genres = $this->getGenres($m->idMovie);
            $m->actors = $this->getActors($m->idMovie);
        }

        return $movies;
    }

```

```

    }

    public function getPaginationCount($dateArray){
        $prodNum = \DB::table("movie as m")
            ->join("projection as p", "m.id", "p.movie_id")
            ->where([
                ["p.starts_at", "<", $dateArray["plus1Day"]],
                ["p.starts_at", ">", $dateArray["minus1Day"]],
                ["m.is_deleted", 0],
                ["p.is_deleted", 0],
                ["m.in_cinema", 1]
            ])
            ->select("m.id")
            ->groupBy("m.id")
            ->get();
        return ceil(count($prodNum) / $this->limit);
    }

    public function getMovies($dateArray, $offset = 0){ //, $differentLimit = 0
        // $dateArray = $this->getDate();
        $offset = $offset * $this->limit;
        $movies = \DB::table("movie AS m")
            ->join("movie_picture AS mp", "m.id", "mp.movie_id")
            ->join("picture AS p", "mp.picture_id", "p.id")
            ->join("projection as pro", "pro.movie_id", "m.id")
            ->select("m.id AS idMovie", "m.description", "m.name AS
movieName", "m.running_time", "m.age_limit", "m.description", "m.in_cinemas_from", "p.p
ath", "p.name AS picName", "p.alt")
            ->where([
                ["pro.starts_at", "<", $dateArray["plus1Day"]],
                ["pro.starts_at", ">", $dateArray["minus1Day"]],
                ["pro.is_deleted", 0],
                ["m.is_deleted", 0],
                ["m.in_cinema", 1]
            ]);
        $movies = $movies->offset($offset)->limit($this->limit)
            ->groupBy("idMovie", "m.description", "movieName", "m.running_time", "m.age_limit", "m.d
escription", "m.in_cinemas_from", "p.path", "picName", "p.alt" )
            ->orderBy("m.name", "ASC")
            ->get();

        foreach($movies as $m){
            $m->grade = $this->getGrade($m->idMovie);
            $m->projections = $this->getProjections($m->idMovie, $dateArray);
            $m->genres = $this->getGenres($m->idMovie);
        }

        return $movies;
    }

    public function getGrade($idMovie){
        $gradeString = \DB::table("user_movie_grade AS umg")
            ->select(\DB::raw("(SUM(grade)/COUNT(*)) AS grade"))
            ->where("umg.movie_id", $idMovie)
            ->groupBy("umg.movie_id")
            ->first();
        if($gradeString){
            return number_format($gradeString->grade, 1);
        }
    }

```

```

    }
    return number_format("0", 1);
}

public function getProjections($idMovie, $dateArray, $technology = 0){
    $whereArray = [];
    array_push($whereArray,
["pr.movie_id",$idMovie],["pr.starts_at","<",$dateArray["plus1Day"]],["pr.starts_a
t",">",$dateArray["minus1Day"]],
    ["pr.is_deleted", 0]);
    if($technology != 0){
        array_push($whereArray, ["t.id",$technology]);
    }
    $projections = \DB::table("projection AS pr")
        ->join("hall AS h","pr.hall_id","h.id")
        ->join("movie_technology AS mt","pr.movie_technology_id", "mt.id")
        ->join("technology AS t","mt.technology_id","t.id")
        ->where($whereArray)
        ->select("pr.id","pr.starts_at","t.name AS
technologyName","pr.reservation_available","h.name")
        ->orderBy("pr.starts_at")
        ->get();

    foreach($projections as $p){
        if($p->starts_at <= $dateArray["dateTimeNow"]){
            $p->projectionAlreadyStartedEnded = 1;
        }else{
            $p->projectionAlreadyStartedEnded = 0;
        }
    }
    return $projections;
}

public function getGenres($idMovie){
    return \DB::table("genre AS g")
        ->join("movie_genre AS mg","g.id","mg.genre_id")
        ->where("mg.movie_id",$idMovie)
        ->select("g.genre","g.id")
        ->get();
}

public function getActors($idMovie){
    return \DB::table("actor AS a")
        ->join("movie_actor AS ma","a.id","ma.actor_id")
        ->where("ma.movie_id",$idMovie)
        ->select("a.name","a.id")
        ->get();
}

public function getMovie($id){
    $movie = \DB::table("movie AS m")
        ->join("movie_picture AS mp","m.id","mp.movie_id")
        ->join("picture AS p","mp.picture_id","p.id")
//        ->join("projection as pro","pro.movie_id","m.id")
        ->join("director AS d","m.director_id","d.id")
        ->join("distributer AS di","m.distributer_id","di.id")
        ->join("country as c","m.country_id","c.id")
        ->select("c.name AS country","di.name AS distributer","d.name AS
director","m.id AS idMovie","m.description_detailed","m.name AS

```

```

movieName", "m.running_time", "m.age_limit", "m.description", "m.in_cinemas_from", "p.p
ath", "p.name AS picName", "p.alt")//, "p1.name as sliderPicName"
    ->where([
        ["m.id", $id],
//        ["pro.starts_at", "<", $dateArray["plus1Day"]],
//        ["pro.starts_at", ">", $dateArray["minus1Day"]]
    ])
    -
>groupBy("c.name", "di.name", "d.name", "idMovie", "m.description_detailed", "movieName
", "m.running_time", "m.age_limit", "m.description", "m.in_cinemas_from", "p.path", "pic
Name", "p.alt")//, "p1.name"
    ->first();

//      $movie->projections = $this->getProjections($movie->idMovie,
$dateArray);
    $movie->actors = $this->getActors($movie->idMovie);
    $movie->grade = $this->getGrade($movie->idMovie);
    $movie->genres = $this->getGenres($movie->idMovie);
    $movie->comments = $this->getComments($movie->idMovie);
    $movie->commentsCount = $this->getNumOfComments($movie->idMovie);
    $movie->backgroundPicName = $this->getBackgroundPicName($movie->idMovie);

    return $movie;
}

public function getBackgroundPicName($idMovie){
    return \DB::table("picture AS p")
        ->join("movie_picture AS mp", "p.id", "mp.about_movie_pic_id")
        ->where("mp.movie_id", $idMovie)
        ->select("p.name", "p.path")
        ->first();
}

public function getComments($idMovie){
    $com = \DB::table("comment AS c")
        ->join("user AS u", "c.user_id", "u.id")
        ->join("movie AS m", "c.movie_id", "m.id")
        ->select("c.movie_id", "c.id", "c.text", "c.inserted_at AS
posted_at", "c.updated_at", "u.username", "u.gender_id", "u.registered_at", "c.user_id"
, "m.id AS idMovie")
        ->where([
            ["m.id", $idMovie],
            ["c.is_deleted", 0],
            ["c.parent_id", NULL]
        ])
        ->offset(0)
        ->limit($this->limitComments)
        ->orderBy("c.inserted_at", "desc")
        ->get();
    return $com;
}

public function getNumOfComments($idMovie){
    return \DB::table("comment")
        ->where([
            ["movie_id", $idMovie],
            ["is_deleted", 0],
            ["parent_id", NULL]
        ])
}

```

```

        ->count();
    }

    public function getMoviesDdl($dateArray){

        $query = \DB::table("movie as m")
            ->join("projection as pro","pro.movie_id","m.id")
            ->select("m.id", "m.name");
        $query = $query->where([
            ["pro.starts_at", "<", $dateArray["plus1Day"]],
            ["pro.starts_at", ">", $dateArray["dateTimeNow"]],
            ["pro.is_deleted", 0],
            ["pro.reservation_available", 1],
            ["m.is_deleted", 0],
            ["m.in_cinema", 1]
        ]);
        $movies = $query
            ->groupBy("m.id", "m.name")
            ->get();
        return $movies;
    }

    public function comingSoon(){
        $cs = \DB::table("movie AS m")
            ->join("movie_picture AS mp","m.id","mp.movie_id")
            ->join("picture AS p","mp.picture_id","p.id")
            ->select("m.id AS idMovie", "m.description", "m.name AS
movieName", "m.running_time", "m.in_cinemas_from", "p.path", "p.name AS
picName", "p.alt")
            ->where([
                ["m.is_deleted", 0],
                ["m.in_cinema", 0]
            ])
            ->get();
        foreach($cs as $m){
            $m->genres = $this->getGenres($m->idMovie);
        }

        return $cs;
    }

    // ***** ADMIN *****

    public function insertMovie($name, $running_time, $desc, $detailed_desc,
$distributer, $director, $country, $date, $actors, $genres, $age_limit, $cs){
        $movieId = \DB::table("movie")
            ->insertGetId([
                "name" => $name,
                "running_time" => $running_time,
                "age_limit" => $age_limit,
                "description" => $desc,
                "description_detailed" => $detailed_desc,
                "in_cinemas_from" => $date,
                "director_id" => $director,
                "country_id" => $country,
                "distributer_id" => $distributer,
                "in_cinema" => $cs,
                "inserted_at" => date("Y-m-d H:i:s"),
            ])
    }

```

```

        "updated_at" => NULL,
        "is_deleted" => 0
    });
//
    dd($movieId);
    \DB::table("movie_actor")
        ->where("movie_id", $movieId)
        ->delete();
    \DB::table("movie_genre")
        ->where("movie_id", $movieId)
        ->delete();

    $this->insertMovieActors($movieId, $actors);
    $this->insertMovieGenres($movieId, $genres);

    return $movieId;
}

public function updateMovie($id, $name, $running_time, $desc, $detailed_desc,
$distributor, $director, $country, $date, $actors, $genres, $age_limit, $cs){
    \DB::table("movie")
        ->where("id", $id)
        ->update([
            "name" => $name,
            "running_time" => $running_time,
            "age_limit" => $age_limit,
            "description" => $desc,
            "description_detailed" => $detailed_desc,
            "in_cinemas_from" => $date,
            "director_id" => $director,
            "country_id" => $country,
            "distributor_id" => $distributor,
            "in_cinema" => $cs,
            "updated_at" => date("Y-m-d H:i:s")
        ]);
    \DB::table("movie_actor")
        ->where("movie_id", $id)
        ->delete();
    \DB::table("movie_genre")
        ->where("movie_id", $id)
        ->delete();

    $this->insertMovieActors($id, $actors);
    $this->insertMovieGenres($id, $genres);
}

public function deleteMovie($id){
    \DB::table("movie")
        ->where("id", $id)
        ->update([
            "is_deleted" => 1
        ]);
}

public function insertMovieActors($movieId, $actors){
    foreach($actors as $a){
        \DB::table("movie_actor")
            ->insertGetId([
                "id" => NULL,
                "movie_id" => $movieId,

```

```

        "actor_id" => $a
    });
}
}

public function insertMovieGenres($movieId, $genres){
    foreach($genres as $g){
        \DB::table("movie_genre")
            ->insertGetId([
                "id" => NULL,
                "movie_id" => $movieId,
                "genre_id" => $g
            ]);
    }
}

public function getMovieData($id){
    $movie = \DB::table("movie AS m")
        ->join("movie_picture AS mp", "m.id", "mp.movie_id")
        ->select("m.id AS
idMovie", "m.description", "m.description_detailed", "m.name AS
movieName", "m.running_time", "m.director_id",
"m.distributer_id", "m.country_id", "m.in_cinemas_from", "age_limit", "in_cinema",
"mp.is_active_slider")
        ->where([
            ["m.id", $id],
        ])
        ->first();
    $movie->actors = $this->getActors($movie->idMovie);
    $movie->genres = $this->getGenres($movie->idMovie);
    $movie->technologies = $this->getMovieTechnologies($movie->idMovie);
    return $movie;
}

public function getMovieTechnologies($idMovie){
    return \DB::table("movie AS m")
        ->join("movie_technology AS mt", "m.id", "mt.movie_id")
        ->join("technology AS t", "mt.technology_id", "t.id")
        ->select("t.id", "t.name")
        ->where("m.id", $idMovie)
        ->get();
}

public function getMoviesAdmin($offset, $searched=0){
    $offset = $offset * $this->adminLimit;
    $query = [["m.is_deleted", 0], ["m.in_cinema", 1]];
    if($searched){
        $query[] = ["m.name", "LIKE", "%$searched%"];
    }
    $movies = \DB::table("movie AS m")
        ->join("director AS d", "m.director_id", "d.id")
        ->join("movie_picture AS mp", "m.id", "mp.movie_id")
        ->join("picture AS p", "mp.picture_id", "p.id")
        ->select("m.id AS idMovie", "m.description", "m.name AS
movieName", "m.running_time", "m.age_limit", "m.description", "m.in_cinemas_from", "p.p
ath", "p.name AS picName", "p.alt", "d.name AS dirName")
        ->where($query);
    $movies = $movies->offset($offset)->limit($this->adminLimit)

```

```

-
>groupBy("idMovie", "m.description", "movieName", "m.running_time", "m.age_limit", "m.d
escription", "m.in_cinemas_from", "p.path", "picName", "p.alt", "d.name")
->get();
    foreach($movies as $m){
        $m->grade = $this->getGrade($m->idMovie);
        $m->genres = $this->getGenres($m->idMovie);
        $m->actors = $this->getActors($m->idMovie);
    }
//    dd($movies);
    return $movies;
}

public function getAllMoviesAdmin(){
    return \DB::table("movie AS m")
        ->where([
            ["m.is_deleted", 0],
            ["m.in_cinema", 1]
        ])
        ->get();
}

public function getAdminPaginationCount(){
    $prodNum = \DB::table("movie as m")
        ->where([
            ["m.is_deleted", 0],
            ["m.in_cinema", 1]
        ])
        ->count();
    return ceil($prodNum / $this->adminLimit);
}

public function insertImage($fileName, $alt, $type, $folder){
    return \DB::table("picture")
        ->insertGetId([
            "name" => $fileName,
            "alt" => $alt,
            "type" => $type,
            "path" => $folder
        ]);
}

public function insertMovieImages($movieId, $imagesArray, $activeSlider){
    $queryArray = $imagesArray;
    $queryArray["id"] = NULL;
    $queryArray["is_active_slider"] = $activeSlider;
    $queryArray["movie_id"] = $movieId;
//    dd($queryArray);
    \DB::table("movie_picture")
        ->insertGetId($queryArray);
}

public function updateMovieImages($movieId, $imagesArray, $activeSlider){
    $queryArray = $imagesArray;
    $queryArray["is_active_slider"] = $activeSlider;
    \DB::table("movie_picture")
        ->where("movie_id", $movieId)
        ->update($queryArray);
}

```



```
}
```

4.3.16 PageSectionModel.php

```
<?php

namespace App\Models;

class PageSectionModel
{
    public function getPageSection($table){
        return \DB::table($table)
            ->get();
    }
}
```

4.3.17 Price

```
<?php

namespace App\Models;

class Price
{
    public function getPrices(){
        return \DB::table("price_weekday_technology as pwt")
            ->join("headline as h","pwt.headline_id", "h.id")
            ->select("h.text","pwt.price")
            ->groupBy("h.text","pwt.price")
            ->get();
    }
}
```

4.3.18 Projection.php

```
<?php

namespace App\Models;

class Projection
{
    public function deleteProjection($id){
        return \DB::table("projection")
            ->where("id", $id)
            ->update([
                "is_deleted" => 1
            ]);
    }
}
```

```

public function getProjections($searched = 0){
    $where = [["p.is_deleted", 0]];
    if($searched){
        $searched = "%$searched%";
        $where[] = ["m.name", "LIKE", $searched];
    }
    return \DB::table("projection AS p")
        ->join("movie AS m", "p.movie_id", "m.id")
        ->join("movie_technology AS mt", "mt.movie_id", "m.id")
        ->join("technology AS t", "mt.technology_id", "t.id")
        ->join("hall AS h", "p.hall_id", "h.id")
        ->where($where)
        ->select("m.name", "p.*", "t.name AS technName", "h.name AS hallName")
        ->orderBy("m.name", "ASC")
        ->orderBy("p.starts_at", "ASC")
        ->get();
}

public function updateProjection($idProjection, $movie, $dateTimeProj, $hall,
$technology, $reservation){
    return \DB::table("projection")
        ->where("id", $idProjection)
        ->update([
            "movie_id" => $movie,
            "starts_at" => $dateTimeProj,
            "hall_id" => $hall,
            "movie_technology_id" => $technology,
            "reservation_available" => $reservation
        ]);
}

public function insertProjections($projectionMovie, $dateTimeProj,
$projectionHall, $projectionTechnology, $projectionResAvailable){
    return \DB::table("projection")
        ->insertGetId([
            "id" => NULL,
            "movie_id" => $projectionMovie,
            "starts_at" => $dateTimeProj,
            "hall_id" => $projectionHall,
            "movie_technology_id" => $projectionTechnology,
            "reservation_available" => $projectionResAvailable,
            "is_deleted" => 0
        ]);
}

public function projectionAvailable($projectionHall, $dateArray, $idProjection
= 0){
    $query = [ ["p.hall_id", $projectionHall], ["p.starts_at", ">",
$dateArray["minus1Day"]], ["p.starts_at",
"<", $dateArray["plus1Day"]], ["p.is_deleted", 0]];
    if($idProjection){
        $query[] = ["p.id", "<>", $idProjection];
    }

    return \DB::table("projection AS p")
        ->join("movie AS m", "p.movie_id", "m.id")
        ->select(\DB::raw("(p.starts_at + INTERVAL m.running_time MINUTE +
INTERVAL 30 MINUTE) as hallFreeAt, p.starts_at, m.running_time"))

```

```

        ->where($query)
        ->get();
    }

    public function getProjectionsDdl($dateArray, $movieId){
        $whereArray = [];
        array_push($whereArray,
["pr.movie_id",$movieId],["pr.starts_at","<",$dateArray["plus1Day"]],
["pr.starts_at",">",$dateArray["minus1Day"]],["pr.reservation_available",">","0"],["pr.starts_at", ">",$dateArray["dateTimeNow"]],
["pr.is_deleted", 0]);

        $projections = \DB::table("projection AS pr")
        ->join("movie_technology AS mt","pr.movie_technology_id", "mt.id")
        ->join("technology AS t","mt.technology_id","t.id")
        ->where($whereArray)
        ->select("pr.id","pr.starts_at","t.name AS technologyName")
        ->orderBy("pr.starts_at")
        ->get();

        return $projections;
    }

    public function getProjection($projectionId){//$movieId,
        return \DB::table("projection as p")
        ->join("movie as m","p.movie_id","m.id")
        ->join("hall as h","p.hall_id","h.id")
        ->join("movie_technology as mt", "p.movie_technology_id","mt.id")
        ->join("technology as t","mt.technology_id","t.id")
        ->select("m.id as idMovie","m.name as movieName","p.id as
idProjection","p.starts_at","t.name as technName","mt.id as idTechnology",
        "h.name as hallName", "h.id as idHall","p.reservation_available")
        ->where([
            ["p.id",$projectionId],
            ["m.id", $movieId]
        ])
        ->first();
    }
}

```

4.3.19 Reservation.php

```

<?php

namespace App\Models;

class Reservation
{
    public function makeReservation($seatsNum, $projectionId,$userId, $todayDate,
$seatIdArray){
        $code = 200;
        try{
            $success = \DB::table("reservation")
                ->insertGetId([
                    "id" => NULL,

```

```

        "user_id" => $userId,
        "projection_id" => $projectionId,
        "reserved_at" => $todayDate,
        "tickets_number" => $seatsNum
    ]);
    if($success){
        foreach($seatIdArray as $s){
            $success2 = \DB::table("reservation_seat")
                ->insertGetId([
                    "id" => NULL,
                    "seat_id" => $s,
                    "reservation_id" => $success
                ]);
            if(!$success2){
                $code = 409;
                break;
            }
        }
        $code = 201;
    }else{
        $code = 409;
    }
}
} catch(\PDOException $ex){
    $code = 409;
    //dd($ex->getMessage());
}
return $code;
}

public function calculatePriceReservation($dayOfWeek, $idTechnology){
    return \DB::table("price_weekday_technology as pwt")
        ->select("price")
        ->where([
            ["pwt.weekday_id", $dayOfWeek],
            ["pwt.technology_id", $idTechnology]
        ])
        ->first();
}

public function getHallRows($idHall, $projectionId){
    $rows = \DB::table("row as r")
        ->where("r.hall_id", $idHall)
        ->orderBy("r.id", "desc")
        ->get();
    foreach($rows as $r){
        $r->seats = $this->getRowSeats($r->id, $projectionId);
    }
    return $rows;
}

public function getRowSeats($rowId, $projectionId){
    $seats = \DB::table("seat as s")
        ->where("s.row_id", $rowId)
        ->get();
    foreach($seats as $s){
        $s->is_available = $this->isSeatAvailable($projectionId, $s->id);
    }
}

```

```

        return $seats;
    }

    public function isSeatAvailable($projectionId, $seatId){
        $exists = \DB::table("reservation as r")
            ->join("reservation_seat as rs","r.id","rs.reservation_id")
            ->where([
                ["r.projection_id",$projectionId],
                ["rs.seat_id", $seatId]
            ])
            ->first();
        if($exists){
            return 0;
        }
        return 1;
    }
}

```

4.3.20 Role.php

```

<?php

namespace App\Models;

class Role
{
    public function getRoles(){
        return \DB::table("role")
            ->get();
    }
}

```

4.3.21 Technology.php

```

<?php

namespace App\Models;

class Technology
{
    public function deleteTechnologyMovie($id){
        \DB::table("movie_technology")
            ->where("movie_id", $id)
            ->delete();
    }

    public function insertTechnologyMovie($technology, $movieId){
        \DB::table("movie_technology")
            ->insertGetId([
                "id" => NULL,
                "movie_id" => $movieId,
                "technology_id" => $technology
            ]);
    }
}

```

```

    }

    public function getMovieTechnologiesProjection($id){
        return \DB::table("movie_technology AS mt")
            ->join("technology AS t", "mt.technology_id", "t.id")
            ->join("projection AS p", "mt.movie_id", "p.movie_id")
            ->select("mt.id", "t.name")
            ->where("p.id", $id)
            ->get();
    }

    public function getMovieTechnologies($movieId){
        return \DB::table("movie_technology AS mt")
            ->join("technology AS t", "mt.technology_id", "t.id")
            ->select("mt.id", "t.name")
            ->where("movie_id", $movieId)
            ->get();
    }

    public function getTechnologies(){
        return \DB::table("technology")
            ->get();
    }
}

```

4.3.22 User.php

```

<?php

namespace App\Models;

class User
{
    public function deleteUser($id){
        \DB::table("user")
            ->where("id", $id)
            ->update([
                "is_deleted" => 1
            ]);
    }

    public function updateUser($id, $username, $password, $email, $gender, $role, $news){
        $query = ["username" => $username,
            "email" => $email,
            "gender_id" => $gender,
            "role_id" => $role,
            "send_via_mail" => $news,
            "updated_at" => date("Y-m-d H:i:s")];
        if($password){
            $query["password"] = md5($password);
        }
        // dd($query);
        return \DB::table("user")
            ->where("id", $id)
            ->update($query);
    }
}

```

```

}

public function getUser($id){
    return \DB::table("user")
        ->where("id",$id)
        ->get();
}

public function getUsers(){
    return \DB::table("user AS u")
        ->join("role AS r","u.role_id", "r.id")
        ->join("user_genre AS g", "u.gender_id", "g.id")
        ->select("u.id AS idUser","u.username","u.email","u.password",
"u.send_via_mail", "r.role AS roleName", "g.name AS genderName")
        ->where("u.is_deleted", 0)
        ->get();
}

public function changePassword($email, $newPass){
    \DB::table("user")
        ->where("email",$email)
        ->update([
            "password" => md5($newPass)
        ]);
}

public function getUsername($email){
    return \DB::table("user")
        ->select("username")
        ->where([
            ["email", "=", $email]
        ])
        ->first();
}

public function getUserByUsernamePassword($username, $password){
    return \DB::table("user")
        ->select("username", "role_id", "id", "email", "gender_id",
"registered_at")
        ->where([
            ["password", "=", md5($password)],
            ["username", "=", $username],
            ["is_deleted", 0]
        ])
        ->first();
}

public function registerUser($username, $password, $tel, $email,
$selectedGender, $sendViaMail){
    date_default_timezone_set('Europe/Belgrade');
    try{
        $insertedId = \DB::table('user')
            ->insertGetId(
                ['id' => NULL, 'username' => $username, 'email' => $email,
'password' => md5($password),
                'registered_at' => date('Y-m-d H:i:s'), 'role_id' =>
1, 'send_via_mail' => intval($sendViaMail),
                'gender_id' => $selectedGender, 'inserted_at' => date('Y-
m-d H:i:s'), 'updated_at' => NULL, 'is_deleted' => 0]
            );
    }
}

```

```

        );

        if($insertedId){
            $code = 201;
        }else{
            $code = 409;
        }
    }catch(\PDOException $ex) {
        $code = 409;
    }
    return $code;
}
}

```

4.4 Requests

4.4.1 InsertMovieRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class InsertMovieRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "picMovIns" => "required|mimes:jpeg,jpg,png",
            "picMovBgIns" => "required|mimes:jpeg,jpg,png",
            "picSliderMovIns" => "nullable|mimes:jpeg,jpg,png"
        ];
    }
}
// |mimes:jpeg,jpg,png

public function messages(){
    return[
        "picMovIns.mimes" => "Picture must be jpeg, jpg or png.",
        "picMovIns.required" => "Movie poster is required.",
        "picMovBgIns.mimes" => "Picture must be jpeg, jpg or png.",
        "picMovBgIns.required" => "Background movie picture is required.",
        "picSliderMovIns.mimes" => "Picture must be jpeg, jpg or png."
    ];
}

```



```
}  
}
```

4.4.2 InsertProjectionsRequest.php

```
<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class InsertProjectionsRequest extends FormRequest  
{  
    /**  
     * Determine if the user is authorized to make this request.  
     *  
     * @return bool  
     */  
    public function authorize()  
    {  
        return true;  
    }  
  
    /**  
     * Get the validation rules that apply to the request.  
     *  
     * @return array  
     */  
    public function rules()  
    {  
        $rules = [];  
  
        foreach($this->request->get('projectionDate') as $key => $val)  
        {  
            $rules['projectionDate.'.$key] = 'after:yesterday';  
        }  
  
        return $rules;  
    }  
  
    public function messages(){  
        $messages = [];  
        foreach($this->request->get('projectionDate') as $key => $val)  
        {  
            $messages['projectionDate.'.$key.'.after'] = "Date can't be in the  
past.";  
        }  
        return $messages;  
    }  
}
```

4.4.3 LoginRequest.php

```
<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;
```

```

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "logUsername" => "required|regex:/^[A-Za-z]{6,15}[\d]{1,5}$/",
            "logPass" => "required|regex:/[\d\w@!\$%^*~\-\.\_\!\\+\=\)\(\{\}\}\?\/\<\>]{6,13}/"
        ];
    }

    public function messages(){
        return [
            "logUsername.required" => "Enter your username.",
            "logUsername.regex" => "Username must contain at least 6 characters and at least 1 number.",
            "logPass.required" => "Enter your password.",
            "logPass.regex" => "Password must contain at least 6 characters."
        ];
    }
}

```

4.4.4 MembershipRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class MembershipRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**

```

```

    * Get the validation rules that apply to the request.
    *
    * @return array
    */
    public function rules()
    {
        return [
            "memName" => "required|regex:/^[A-Za-z]{6,15}(\s[A-Za-z]{6,15})*$/",
            "memLastname" => "required|regex:/^[A-Za-z]{6,15}(\s[A-Za-z]{6,15})*$/",
            "memEmail" => "required|email"
        ];
    }

    public function messages(){
        return [
            "memName.required" => "Enter your firstname.",
            "memName.regex" => "Firstname is not in valid format.",
            "memLastname.required" => "Enter your lastname.",
            "memLastname.regex" => "Lastname is not in valid format.",
            "memEmail.required" => "Enter your email.",
            "memEmail.regex" => "Email is not in valid format"
        ];
    }
}

```

4.4.5 RegistrationRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegistrationRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "username" => "required|regex: /^[A-Za-z]{6,15}[\d]{1,5}$/",
            "pass" => "required|regex: /[\d\w\@$%^*~\-\.\_\!\\+=\)\(\{\}\}\?\\/\<\>]{6,13}/",
            "pass1" => "required|same:pass",
            "email" => "required|email",

```

```

        "tel" => "nullable|regex:/^06[\d]\-[\d]{3}\-[\d]{3,4}$/",
        "selectedGender" => "required",
        "sendViaMail" => "nullable"
    ];
}

public function messages(){
    return [
        "username.required" => "Enter your username.",
        "username.regex" => "Username must contain at least 6 characters and
at least 1 number.",
        "pass.required" => "Enter your password.",
        "pass.regex" => "Password must contain at least 6 characters.",
        "pass1.required" => "Repeat your password.",
        "pass1.same" => "Invalid repeated password.",
        "email.required" => "Email is required.",
        "email.email" => "Email is not in valid format.",
        "tel.regex" => "Invalid phone number format.",
        "selectedGender.required" => "Choose gender."
    ];
}
}

```

4.4.6 UpdateMovieRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class UpdateMovieRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "picMovUpd" => "nullable|mimes:jpeg,jpg,png",
            "picMovBgUpd" => "nullable|mimes:jpeg,jpg,png",
            "picSliderMovUpd" => "nullable|mimes:jpeg,jpg,png"
        ];
    }

    public function messages(){
        return[

```

```

        "picMovUpd.mimes" => "Picture must be jpeg, jpg or png.",
        "picMovBgUpd.mimes" => "Picture must be jpeg, jpg or png.",
        "picSliderMovUpd.mimes" => "Picture must be jpeg, jpg or png."
    ];
}
}

```

4.4.7 UpdateProjectionRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class UpdateProjectionRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "dateProjectionUpdate" => 'after:yesterday'
        ];
    }

    public function messages(){
        return [
            "dateProjectionUpdate.after" => "Date can't be in the past."
        ];
    }
}

```

4.5 Middleware

4.5.1 IsAdminLoggedIn

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsAdminLoggedIn

```

```

{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle(Request $request, Closure $next)
    {
        if($request->session()->has("user")){
            $user = $request->session()->get("user");
            if($user->role_id != 2){
                return redirect("/home");
            }else{
                return $next($request);
            }
        }else{
            return redirect("/home");
        }
    }
}

```

4.5.2 IsLoggedIn.php

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsLoggedIn
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has("user")){
            return $next($request);
        }else{
            return redirect("/home");
        }
    }
}

```

4.6 JS

4.6.1 main.js

```

window.onload = function(){
    var url = window.location.href;
    if(url.indexOf("home") !== -1){

```

```

        console.log("dfg");

    }else if(url.indexOf("admin") !== -1){

        $("#top").click(function(){
            $("html,body").animate({ scrollTop: 0 }, 1000);
        });
        $(window).scroll(function(){
            if($(this).scrollTop()>170){
                $("#top").fadeIn();
            }else{
                $("#top").fadeOut();
            }
        });
        $("#menuAdmin a").on('click', function(e) {
            e.preventDefault();
            $('#menuAdmin li').removeClass('active');
            $(this).parent().addClass('active');
            var target = $(this).attr('href');
            $('html, body').animate({
                scrollTop: ($(target).offset().top)
            }, 2000);
        });
        $("#activityList").change(sortActivity);
        $("#insLink").click(checkInsert);
        $("#closeInsertLinkForm").click(function(){
            $('#blurBgInsertLink').fadeOut();
            $(document).unbind('scroll');
            $('body').css({'overflow':'visible'});
        });
        $("#linksAdmin #insertLink").click(function(){
            $('#blurBgInsertLink').fadeIn();
            $(document).bind('scroll');
            $('body').css({'overflow':'hidden'});
        });
        $("#closeUpdateLinkForm").click(function(){
            $('#blurBgUpdateLink').fadeOut();
            $(document).unbind('scroll');
            $('body').css({'overflow':'visible'});
        });
        $("#updLink").click(updateLink);
        $("#linksTable .fa-edit").click(editLinkClick);
        $("#commentsHolderAdmin .fa-remove").click(deleteAdminComment);
        $("#updUser").click(checkUpdateUser);
        $("#blurBgUpdateUser #closeUpdateUserForm").click(function(){
            $('#blurBgUpdateUser').fadeOut();
            $(document).unbind('scroll');
            $('body').css({'overflow':'visible'});
        });
        $("#tableUsers .fa-edit").click(getUser);
        $("#updateProjForm").submit(checkProjectionUpd);
        $("#searchProjections").keyup(searchProjectionsByMovie);
        $("#closeUpdateProjForm").click(function(){
            $('#blurUpdateProj').fadeOut();
            $(document).unbind('scroll');
            $('body').css({'overflow':'visible'});
        });
        $(".prInfo .fa-edit").click(editProjClick);
    }

```

```

    $("#seeProjections").click(function(e){
        e.preventDefault();
        $("#viewProjections").toggle();
        if($("#viewProjections").is(":visible")){
            $(this).html("View less <span class='fa fa-hand-o-up'></span>");
        }else{
            $(this).html("View more <span class='fa fa-hand-o-down'></span>");
        }
    });
    $("body").on("change", ".projectionMovie", function(){
        getMovieTechnologies($(this));
    });
    $("#projectionsForm").submit(checkProjectionsInsert);
    $("body").on("click", ".adminProjection .projectionX .fa-close",function(){
        $(this).parent().parent().remove();
        let num = $("#numOfProjections").val();
        $("#numOfProjections").val(num - 1);
    });
    $("#numOfProjections").change(printMoreProjections);
    // $("#btnDeleteMov").click(deleteMovie);
    $("#genresMovIns").change(addMovieGenreInsert);
    $("#actorsMovIns").change(addMovieActorInsert);
    $("#insertMovieForm").submit(checkMovieInsert);
    $("#closeInsertMovieForm").click(function(){
        $('#blurBgInsertMovie').fadeOut();
        $(document).unbind('scroll');
        $('body').css({'overflow':'visible'});
    })
    $("#insertMovie").click(function(){
        $('#blurBgInsertMovie').fadeIn();
        $('body').css({'overflow':'hidden'});
        $(document).bind('scroll',function () {
            window.scrollTo(0,0);
        });
    });
    $("#updateMovieForm").submit(checkMovieUpdate);
    $("body").on("click", ".movUpdActGenres .fa-close",function(){
        $(this).parent().remove();
    });
    $("#genresMovUpd").change(addMovieGenre);
    $("#actorsMovUpd").change(addMovieActor);
    $(".btnUpdateMov").click(getMovieData);
    $("#closeUpdateMovieForm").click(function(){
        $('#blurBgUpdateMovie').fadeOut();
        $(document).unbind('scroll');
        $('body').css({'overflow':'visible'});
    });
    $("body").on("click", ".paginationAdmMov",function(e){
        e.preventDefault();

        let offset = $(this).data("limit");
        let searched = $("#extendSearchAdm").val().trim();
        if(searched == ""){
            searched = 0;
        }
        $('.paginationAdmMov').removeClass('activePagination');
        $(this).addClass('activePagination');
    });

```



```

        $.ajax({
            url: "admin/moviesPage/" + offset + "/" + searched , //
            "api/home/" + offset
            method: "GET",
            dataType: 'json',
            success: function(data){
                printMoviesAdmin(data.movies);
                $("html, body").animate({ scrollTop:
$('#adminMovies').position().top }, 1000);
            },
            error: function(error){
                console.log(error);
            }
        });
    });
    $("#extendSearchAdm").focus();
    $("#extendSearchAdm").keyup(function(event){
        let searched = $("#extendSearchAdm").val().trim();
        searchMoviesAdm(searched);
    });

}
$("#membershipRequestForm").submit(becomeMemberRequest);
$("#contactAdmin").submit(contactAdmin);
$("#contactCinema").submit(contactCinema);
$("#accountAccessForm").submit(checkAccountAccessForm);
$("#forgetPassForm").submit(checkForgetPassEmail);
$("#resetPass").click(resetPassClicked);
$(".voteDiv .confirmVote").click(sendVote);
$("body").on("click", ".giveVoteList .fa", function(){
    rateStarClicked($(this));
});
$(".faCloseRateCloud").click(closeRateDiv);
$(".vote").click(showVoteDiv);
// $("#abtMovieBook").click(scrollToMovieProj);
$("#inputSeatsNum").change(changeSeatsNum);
$("#confirmReservation").click(makeReservation);
$("body").on("click", "#seatsChosenList .fa-close", function(){
    $(this).parent().remove();
    let moreSeats = $("#selectMoreSeats span").text();
    let numMoreSeats = parseInt(moreSeats);
    numMoreSeats++;
    $("#selectMoreSeats span").text(numMoreSeats);
});
$(".available_seat").click(chooseSeat);

$("#menuList a[href$='reservation']").click(showReservation);
$("#help").click(viewPrices);

$('#closeForgetPassForm').click(function(){
    $('#blurBackgroundForgetPass').fadeOut();
    $(document).unbind('scroll');
    $('body').css({'overflow':'visible'});
});
$("#closePrices").click(function(){
    $('#blurBackgroundPrices').fadeOut();
    $(document).unbind('scroll');
    $('body').css({'overflow':'visible'});
});

```

```

$("#closeResForm").click(function(){
    $('#blurBackgroundReservation').fadeOut();
    $(document).unbind('scroll');
    $('body').css({'overflow':'visible'});
});

makeDdl();
/* If the user clicks anywhere outside the select box,
then close all select boxes: */
document.addEventListener("click", closeAllSelect);

$("#viewMoreCom").click(viewMoreComments);
$("body").on("click", "#comments .fa-close", function(e){
    let currentTextInTa = $(this).parent().find('.updateCommArea').val();
    $(this).parent().find('.fa-edit').css("display","initial");
    let html = `<div class="commText">${currentTextInTa}</div>`;
    $(this).parent().find('.commentTextDiv').html(html);
    $(this).remove();
});
$(".bottomPartComment .fa-edit").click(updateComment);
// $(".bottomPartComment .fa-trash").live("click",function(){
//     $(".bottomPartComment .fa-trash").die("click");
//     deleteComment();
// });
$(".bottomPartComment .fa-trash").click(deleteComment);
$(".bottomPartComment .fa-angle-down").click(showReplies);
$(".bottomPartComment .fa-angle-up").click(hideReplies);
$(".comment .bottomPartComment .reply").click(replyToComment);
$("#myComment").keyup(function(event){
    if (event.keyCode === 13) {
        event.preventDefault();
        $("#leaveComm .fa-pencil").click();
    }
});
$("#leaveComm .fa-pencil").click(postComment);
$('#logFormDiv').hide();
$('#blurBackground').hide();
$('#deleteFiltersDiv').hide();
$("body").on("click", "#deleteFiltersDiv", function(e){
    deleteFilters();
});
$('#logReg #logRegList #logLink').click(function(){
    $('#logFormDiv').slideToggle();
});
$("#logFormX").click(function(){
    $("#logFormDiv").fadeOut();
});
$('#logReg #logRegList #regLink').click(function(){
    $('#blurBackground').fadeIn();
    $('body').css({'overflow':'hidden'});
    $(document).bind('scroll',function () {
        window.scrollTo(0,0);
    });
});

document.querySelector('#btnRegistration').addEventListener("click",checkingRegData);
});
$("#menuList li a").click(showSubmenu);
$("#searchCont .fa-search").click(extendSearch);

```

```

if($("#slider").length){
    movieSlider();
}
$("#movies #extendSearch").keyup(showX);
$("#movies #searchCont .fa-close").click(emptySearchInput);
$(".arrowsSlider").click(changeSliderMovie);

var rbMoviesSort = document.getElementsByName("sortMovies");
for(let rb of rbMoviesSort){
    if(rb.checked){
        rbMoviesSortChecked = rb.value;
    }
}

$(".containerMoviesRadio").click(filterMovies);
$(".ddlFilter").change(filterMovies);
$(".searchicon").click(function(){
    let searched = $("#extendSearch").val();
    if(searched.trim() !== ""){
        filterMovies();
        $("#deleteFiltersDiv").fadeIn();
    }
});
$(".ddlChangingOptionsColor").change(function(){
    let ddlId = $(this).attr("id");
    var index = $(this).prop("selectedIndex");
    let valueSelected = document.getElementById(ddlId)[index].value;
    if(valueSelected !== '0'){
        $(this).removeClass('ddlDefaultOption');
        this.children[0].setAttribute("class", "greyOption");
    }else{
        $(this).addClass('ddlDefaultOption');
        $(this).children().addClass('alwaysWhiteOptionAdded');
        this.children[0].setAttribute("class", "greyOption");
    }
});
$("#reservationForm").submit(continueReservation);
$("#closeRegForm").click(function(){
    $('#blurBackground').fadeOut();
    $(document).unbind('scroll');
    $('body').css({'overflow': 'visible'});
});
$("body").on("click", ".paginationLinks", function(e){
    e.preventDefault();

    let offset = $(this).data("limit");
    $('.paginationLinks').removeClass('activePagination');
    $(this).addClass('activePagination');

    $.ajax({
        url: "page/" + offset , // "api/home/" + offset
        method: "GET",
        dataType: 'json',
        success: function(data){
            printMovies(data.movies);
            $("html, body").animate({ scrollTop: $('#movies').position().top
}, 1000);
        },
        error: function(error){

```

```

        console.log(error);
    }
});

});
}

function becomeMemberRequest(){
    let firstname = $("#memReqName").val();
    let lastname = $("#memReqLastname").val();
    let email = $("#memReqEmail").val();
    let code = $("#memReqCode").val();

    let reName = /^[A-Z][a-z]{3,10}(\s[A-Z][a-z]{3,10})*/$/;
    let reEmail = /^\\w+([\\.-]?\\w+)*\\@\\w+([\\.-]?\\w+)*\\.\\w{2,4}+$/;

    if(email === ""){
        document.querySelector("#memReqEmailErr").innerHTML="Email field is required!";
        errors++;
    }else if(!reEmail.test(email)){
        document.querySelector("#memReqEmailErr").innerHTML="Email is not in valid format.";
        errors++;
    }else{
        document.querySelector("#memReqEmailErr").innerHTML="";
    }

    if(firstname === ""){
        document.querySelector("#memReqNameErr").innerHTML="Firstname field is required!";
        errors++;
    }else if(!reName.test(firstname)){
        document.querySelector("#memReqNameErr").innerHTML="Firstname is not in valid format.";
        errors++;
    }else{
        document.querySelector("#memReqNameErr").innerHTML="";
    }

    if(lastname === ""){
        document.querySelector("#memReqLastnameErr").innerHTML="Lastname field is required!";
        errors++;
    }else if(!reName.test(lastname)){
        document.querySelector("#memReqLastnameErr").innerHTML="Lastname is not in valid format.";
        errors++;
    }else{
        document.querySelector("#memReqLastnameErr").innerHTML="";
    }

    if(code === ""){
        document.querySelector("#memReqCode").innerHTML="Cody field is required!";
        errors++;
    }else{
        document.querySelector("#memReqCode").innerHTML="";
    }
}

```

```

    if(errors){
        return false;
    }else{
        return true;
    }
}

function contactAdmin(){
    let email = $("#emailAddress").val();
    let body = $("#body").val().trim();
    let reEmail=/^\w+([\.-]?\w+)*\@\w+([\.-]?\w+)*(\.\w{2,4})+$/;
    let errors = 0;

    if(email == ""){
        document.querySelector("#emailAddressErr").innerHTML="Email field is required!";
        errors++;
    }else if(!reEmail.test(email)){
        document.querySelector("#emailAddressErr").innerHTML="Email is not in valid format.";
        errors++;
    }else{
        document.querySelector("#emailAddressErr").innerHTML="";
    }

    if(body == ""){
        document.querySelector("#bodyErr").innerHTML="Body cant't be empty.";
        errors++;
    }else{
        document.querySelector("#bodyErr").innerHTML="";
    }

    if(errors){
        return false;
    }else{
        return true;
    }
}

function contactCinema(){
    let firstname = $("#firstnameContCin").val();
    let reName = /^[A-Z][a-z]{3,10}(\s[A-Z][a-z]{3,10})*/;
    let lastname = $("#lastnameContCin").val();
    let email = $("#emailAddressContCin").val();
    let reEmail=/^\w+([\.-]?\w+)*\@\w+([\.-]?\w+)*(\.\w{2,4})+$/;
    let body = $("#question").val();
    let errors = 0;

    if(email === ""){
        document.querySelector("#emailAddressContCinErr").innerHTML="Email field is required!";
        errors++;
    }else if(!reEmail.test(email)){
        document.querySelector("#emailAddressContCinErr").innerHTML="Email is not in valid format.";
        errors++;
    }else{
        document.querySelector("#emailAddressContCinErr").innerHTML="";
    }
}

```

```

    if(firstname === ""){
        document.querySelector("#firstnameContCinErr").innerHTML="Firstname field
is required!";
        errors++;
    }else if(!reName.test(firstname)){
        document.querySelector("#firstnameContCinErr").innerHTML="Firstname is not
in valid format.";
        errors++;
    }else{
        document.querySelector("#firstnameContCinErr").innerHTML="";
    }

    if(lastname === ""){
        document.querySelector("#lastnameContCinErr").innerHTML="Lastname field is
required!";
        errors++;
    }else if(!reName.test(lastname)){
        document.querySelector("#lastnameContCinErr").innerHTML="Lastname is not
in valid format.";
        errors++;
    }else{
        document.querySelector("#lastnameContCinErr").innerHTML="";
    }

    if(body === ""){
        document.querySelector("#questionErr").innerHTML="Body field is
required!";
        errors++;
    }else{
        document.querySelector("#questionErr").innerHTML="";
    }

    if(errors){
        return false;
    }else{
        return true;
    }
}

function checkAccountAccessForm(){
    let email = $("#accountAccessEmail").val();
    let reEmail=/^[^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,4})+$/;
    let pass = $("#accountAccessPass").val();
    let rePass=/[\\d\\w\\@\\$\\^\\*\\-\\.\\_\\!\\+\\=\\)\\(\\{\\}\\}\\?\\|\\<\\>]{6,13}/;
    let code = $("#accountAccessCode").val();
    let errors = 0;

    if(email === ""){
        document.querySelector("#accountAccessEmailErr").innerHTML="Email field is
required!";
        errors++;
    }else if(!reEmail.test(email)){
        document.querySelector("#accountAccessEmailErr").innerHTML="Email is not
in valid format.";
        errors++;
    }else{
        document.querySelector("#accountAccessEmailErr").innerHTML="";
    }
}

```

```

        if(pass === ""){
            document.querySelector("#accountAccessPassErr").innerHTML="Password field
is required!";
            errors++;
        }else if(!rePass.test(pass)){
            document.querySelector("#accountAccessPassErr").innerHTML="Password is not
in valid format.";
            errors++;
        }else{
            document.querySelector("#accountAccessPassErr").innerHTML="";
        }

        if(code === ""){
            document.querySelector("#accountAccessCodeErr").innerHTML="Verification
code field is required!.";
            errors++;
        }else{
            document.querySelector("#accountAccessCodeErr").innerHTML="";
        }

        if(errors){
            return false;
        }else{
            return true;
        }
    }
}

function checkForgetPassEmail(){
    let email = $("#forgetPassEmail").val();
    let reEmail=/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,4})+$/;

    if(email == ""){
        return false;
    }else if(!reEmail.test(email)){
        return false;
    }else{
        return true;
    }
}

function resetPassClicked(e){
    e.preventDefault();
    $('#blurBackgroundForgetPass').fadeIn();
    $('body').css({'overflow':'hidden'});
    $(document).bind('scroll',function () {
        window.scrollTo(0,0);
    });
}

function viewPrices(e){
    e.preventDefault();
    $('#blurBackgroundPrices').fadeIn();
    $('body').css({'overflow':'hidden'});
    $(document).bind('scroll',function () {
        window.scrollTo(0,0);
    });
}

```

```

function sendVote(){
    let idMovie = $(this).data("movid");
    let grade = $(this).parent().find(".userChosenGrade").text();
    let gradeNum =parseFloat(grade);
    let obj = $(this).prev();

    $.ajax({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        },
        url: global_url+"rateMovie",
        method: "post",
        data: {
            grade: gradeNum,
            idMovie: idMovie
        },
        success: function(data,status, xhr){
            let html = "<p class='success'>Thank u for ur vote.</p>";
            obj.next().unbind("click");

            switch(xhr.status) {
                case 204:
                    html = "<p class='success'>Ur vote was updated.</p>";
                    break;
            }
            $(html).insertAfter(obj);
        },
        errors: function(xhr, textStatus, errorThrown){
            switch(xhr.status) {
                // case 404:
                //     error = "Page not found.";
                //     break;
            }
        }
    })
}

function rateStarClicked(obj){
    let index = obj.data('index');

    let html = "";
    for(let i = 0; i < 10; i++){
        if(i <= index){
            html += `<li><span class="fa fa-star" data-index="${i}"></span></li>`;
        }else{
            html += `<li><span class="fa fa-star-o" data-index="${i}"></span></li>`;
        }
    }
    index++;
    let grade = index.toFixed(1);
    obj.parent().parent().parent().find(".userChosenGrade").html(grade);
    obj.parent().parent().html(html);
}

function closeRateDiv(){
    $(this).parent().hide();
}

```



```

function showVoteDiv(e){
    // console.log("ddd");
    e.preventDefault();
    let isVisible = $(this).parent().find(".giveVote").is(':visible');

    if(isVisible){
        $(this).parent().find(".giveVote").hide();
    }else{
        $(this).parent().find(".giveVote").fadeIn();
    }
}

function scrollToMovieProj(e){
    e.preventDefault();
    $("html, body").animate({ scrollTop: $('#projectionsMovie').position().top },
1000);
}

var inputSeatsNum = $("#projectionInfo li:eq(3) span:eq(1)
#inputSeatsNum").length;
if(inputSeatsNum){
    var inputSeatsNumValue = parseInt($("#projectionInfo li:eq(3) span:eq(1)
#inputSeatsNum").val());
}

function changeSeatsNum(e){
    let numberOfChosenSeats = $("#seatsChosenList li").length;
    let value = parseInt($(this).val());
    if(numberOfChosenSeats > value){
        $("#projectionInfo li:eq(3) span:eq(1) #inputSeatsNum").val(value+1);
    }else{
        let pricePerTicket = $("#pricePerTicket").val();
        let decimalPricePerTicket = parseFloat(pricePerTicket);
        let seatsLeftToChoose = $("#selectMoreSeats span").text();
        let seatsLeftToChooseNum = parseInt(seatsLeftToChoose);

        if(value > inputSeatsNumValue){
            seatsLeftToChooseNum++;
        }else{
            if(seatsLeftToChooseNum > 0){
                seatsLeftToChooseNum--;
            }
        }
        inputSeatsNumValue = value;
        let priceSum = (decimalPricePerTicket * inputSeatsNumValue).toFixed(2);
        $("#resSum span").text(priceSum + " $");
        $("#selectMoreSeats span").text(seatsLeftToChooseNum);
    }
}

function makeReservation(){
    let seatsLeftToChoose = $("#selectMoreSeats span").text();
    let seatsLeftToChooseNum = parseInt(seatsLeftToChoose);
    let seats = $("#projectionInfo li:eq(3) span:eq(1) #inputSeatsNum").val();
    let seatsNum = parseInt(seats);
    let projectionId = $("#projectionId").val();
    let seatIdArray = [];
    let lis = document.querySelectorAll("#seatsChosenList li");
    for(let li of lis){

```

```

        seatIdArray.push(parseInt(li.dataset.id));
    }
    // console.log(seatIdArray);

    if(seatsLeftToChooseNum === 0){
        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url: global_url + "makeReservation",
            method: "post",
            data: {
                seatsNum: seatsNum,
                projectionId: projectionId,
                seatIdArray: seatIdArray
            },
            success: function(data){
                $("<p class='success'>Your reservation was successful.
:)</p>").insertAfter("#resSum");
                $("#confirmReservation").unbind("click");
            },
            error: function(err){

            }

        })
    }
}

function chooseSeat() {
    let moreSeats = $("#selectMoreSeats span").text();
    let numMoreSeats = parseInt(moreSeats);

    if (numMoreSeats) {
        numMoreSeats--;
        console.log(numMoreSeats);
        let seat = $(this).data("seat");
        let seatId = $(this).data("id");
        let row = $(this).parent().data("row");
        let html = `<li data-id="${seatId}">Row: ${row}, Seat: ${seat}<span
class="fa fa-close"></span></li>`;
        $("#selectMoreSeats span").text(numMoreSeats);
        $("#seatsChosenList").append(html);
    }
}

function continueReservation(){
    let si_movie = $("#reservationMovieDdl option:selected").index();
    let si_projection = $("#reservationProjectionDdl option:selected").index();

    if(si_movie && si_projection){
        return true;
    }else{
        return false;
    }
}

function makeDdl(printNewDdlFromIndex = -1){

```

```

var x, i, j, l, ll, selElmnt, a, b, c;
i=0;
if(printNewDdlFromIndex > -1){
    i = printNewDdlFromIndex;
}
/* Look for any elements with the class "custom-select": */
x = document.getElementsByClassName("custom-select");
l = x.length;
if(printNewDdlFromIndex > -1){
    l = printNewDdlFromIndex + 1;
}
for (i; i < l; i++) {
    selElmnt = x[i].getElementsByTagName("select")[0];
    ll = selElmnt.length;
    /* For each element, create a new DIV that will act as the selected item:
*/
    if(printNewDdlFromIndex === -1){
        a = document.createElement("DIV");
        a.setAttribute("class", "select-selected");
        a.innerHTML = selElmnt.options[selElmnt.selectedIndex].innerHTML;
        x[i].appendChild(a);
    }
    /* For each element, create a new DIV that will contain the option list:
*/
    b = document.createElement("DIV");
    b.setAttribute("class", "select-items select-hide");
    for (j = 0; j < ll; j++) {
        /* For each option in the original select element,
        create a new DIV that will act as an option item: */
        c = document.createElement("DIV");
        c.innerHTML = selElmnt.options[j].innerHTML;
        c.addEventListener("click", function(e) {
            /* When an item is clicked, update the original select box,
            and the selected item: */
            var y, i, k, s, h, sl, yl;
            s = this.parentNode.parentNode.getElementsByTagName("select")[0];
            sl = s.length;
            h = this.parentNode.previousSibling;
            for (i = 0; i < sl; i++) {
                if (s.options[i].innerHTML == this.innerHTML) {
                    s.selectedIndex = i;
                    // console.log(s);
                    h.innerHTML = this.innerHTML;
                    y = this.parentNode.getElementsByClassName("same-as-
selected");
                    yl = y.length;
                    for (k = 0; k < yl; k++) {
                        y[k].removeAttribute("class");
                    }
                    this.setAttribute("class", "same-as-selected");
                    break;
                }
            }
            h.click();
            s.click();
        });
        b.appendChild(c);
    }
    x[i].appendChild(b);
}

```

```

        if(a){
            a.addEventListener("click", function(e) {
                /* When the select box is clicked, close any other select boxes,
                and open/close the current select box: */
                e.stopPropagation();
                closeAllSelect(this);
                this.nextSibling.classList.toggle("select-hide");
                this.classList.toggle("select-arrow-active");
            });
        }
        if(printNewDdlFromIndex > -1){
            selElmnt.selectedIndex = 0;
            if(printNewDdlFromIndex === 2){
                selElmnt.parentElement.children[1].innerHTML = "Choose
projection..";
            }else{
                selElmnt.parentElement.children[1].innerHTML = "Choose movie..";
            }
            let optionsDivHolder =
selElmnt.parentElement.children[2].getElementsByClassName("same-as-selected");
            let optL = optionsDivHolder.length;
            for (let l = 0; l < optL; l++) {
                optionsDivHolder[l].removeAttribute("class");
            }
            selElmnt.parentElement.children[2].children[0].setAttribute("class",
"same-as-selected");

            let nextList = x[++i];
            if(nextList){
                let selectElement = nextList.getElementsByTagName("select")[0];
                selectElement.innerHTML = "<option value='0'>Choose
projection..</option>";
                selectElement.selectedIndex = 0;
                selectElement.parentElement.children[1].innerHTML = "Choose
projection..";

                let opts =
selectElement.parentElement.children[2].getElementsByTagName("div");
                selectElement.parentElement.children[2].innerHTML = "<div
class='same-as-selected'>Choose projection..</div>";
            }
        }
    }
    $(".custom-select #reservationDateDdl").unbind('click');
    $(".custom-select #reservationMovieDdl").unbind('click');
    $(".custom-select #reservationDateDdl").click(reservationDateDdlChanged);
    $(".custom-select #reservationMovieDdl").click(reservationMovieDdlChanged);
}

function reservationMovieDdlChanged(){
    let si = $(this)[0].selectedIndex;
    let selectedOptionValue = $(this)[0][si].attributes[0].nodeValue;
    let dateIndex = $("#reservationDateDdl")[0].selectedIndex;
    let date = $("#reservationDateDdl
option")[dateIndex].attributes[0].nodeValue;

    if(selectedOptionValue !== '0'){
        updateReservationProjectionList(selectedOptionValue, date);
    }else{

```

```

        let selectElement = document.getElementById("reservationProjectionDdl");
        selectElement.innerHTML = "<option value='0'>Choose
projection..</option>";
        selectElement.selectedIndex = 0;
        selectElement.parentElement.children[1].innerHTML = "Choose projection..";

        let opts =
selectElement.parentElement.children[2].getElementsByTagName("div");
        selectElement.parentElement.children[2].innerHTML = "<div class='same-as-
selected'>Choose projection..</div>";
    }
}

function updateReservationProjectionList(selectedOptionValue, date){
    $.ajax({
        url: global_url + "getProjectionsDdl",
        method: "get",
        data: {
            movieId: selectedOptionValue,
            date: date
        },
        success: function(data){
            let ddl = $("#reservationProjectionDdl");//.find("option").first()
            $("#reservationProjectionDdl").parent().find(".select-
items").remove();

            let html = `<option value='0'>Choose projection..</option>`;
            if(data.projections){
                for(let p of data.projections){
                    let dateObj = new Date(p.starts_at);
                    let dateArray = formatDate(dateObj);
                    html += `<option
value="${p.id}">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}.
${dateArray["hours"]}.${dateArray["minutes"]} (${p.technologyName})</option>`;
                }
            }

            ddl.html(html);

            makeDdl(printNewDdlFromIndex = 2);
        },
        error: function(err){

        }
    })
}

function reservationDateDdlChanged(){
    let si = $(this)[0].selectedIndex;
    let selectedOptionValue = $(this)[0][si].attributes[0].nodeValue;
    //console.log(selectedOptionValue);
    updateReservationMovieList(selectedOptionValue);
}

function updateReservationMovieList(selectedOptionValue){
    $.ajax({
        url: global_url + "getMoviesDdl",
        method: "get",
        data: {

```

```

        date: selectedOptionValue
    },
    success: function(data){
        let ddl = $("#reservationMovieDdl");//.find("option").first()
        $("#reservationMovieDdl").parent().find(".select-items").remove();

        let html = `<option value='0'>Choose movie..</option>`;
        if(data.movies){
            for(let m of data.movies){
                html += `<option value="${m.id}">${m.name}</option>`;
            }
        }

        ddl.html(html);//<option value='0'>Choose
movie..</option><option>dkdkdkdk</option>

        makeDdl(printNewDdlFromIndex = 1);
    },
    error: function(err){

    }
})
}

function closeAllSelect(elmnt) {
    /* A function that will close all select boxes in the document,
    except the current select box: */
    var x, y, i, x1, y1, arrNo = [];
    x = document.getElementsByClassName("select-items");
    y = document.getElementsByClassName("select-selected");
    x1 = x.length;
    y1 = y.length;
    for (i = 0; i < y1; i++) {
        if (elmnt == y[i]) {
            arrNo.push(i)
        } else {
            y[i].classList.remove("select-arrow-active");
        }
    }
    for (i = 0; i < x1; i++) {
        if (arrNo.indexOf(i)) {
            x[i].classList.add("select-hide");
        }
    }
}

function showReservation(e){
    e.preventDefault();
    $('#blurBackgroundReservation').fadeIn();
    $('body').css({'overflow':'hidden'});
    $(document).bind('scroll',function () {
        window.scrollTo(0,0);
    });
}

var global_url = "https://diplomski-movie-blackout.herokuapp.com/";
function hideReplies() {
    $(this).parent().parent().parent().parent().find(".repliesDiv").slideUp();
}

```

```

function showReplies(){
    let replies =
$(this).parent().parent().parent().parent().find(".repliesDiv").find('.replyComment').length;
    let newlyPosted =
$(this).parent().parent().parent().parent().find(".repliesDiv").find('.newlyPosted').length;
    console.log("broj replajeva:" + replies);
    let parent_id = $(this).data('id');
    let repliesDivObj =
$(this).parent().parent().parent().parent().find(".repliesDiv");
    console.log(repliesDivObj);

    if((replies > 0) && (newlyPosted === 0)){
        $(this).parent().parent().parent().parent().find(".replyComment").show();
    }
    $(this).parent().parent().parent().parent().find(".repliesDiv").slideDown();
    }else{
        getRepliesOfComment(parent_id, repliesDivObj);
    }
}

function printReplies(replies, loggedUserId, isNewlyPosted=0){
    let html = ``;
    for(let r of replies){
        html += `<div class="replyComment `;
        if(isNewlyPosted){
            html += `newlyPosted`;
        }
        html += `>`
        <div class="userFaHolder">
            <span class="fa fa-user`;
        if (r.gender_id === 1){
            html += ` femaleUser`;
        }else{
            html += ` maleUser`;
        }
        html += `></span>`
        <p class="memberSince">Member since</p>`;
        let dateObj = new Date(r.registered_at);
        let dateArray = formatDate(dateObj);
        html += `<p
class="memberDate">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}.
</p>`
        </div>
        <div class="commentRight">
        <div class="topPartComment">
            <span class="usernameComment">${r.username}</span>`;
        dateObj = new Date(r.posted_at);
        dateArray = formatDate(dateObj);
        html += `<span
class="commPosted">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}.
${dateArray["hours"]}.${dateArray["minutes"]}</span>`
        </div>
        <div class="commentTextDiv">
            <div class="commText">${r.text}</div>
        </div>
        <div class="bottomPartComment">`;
    }
}

```

```

        if(loggedUserId){
            html += `

```



```

    })
  })
}

function repplyToRepply(){
  let parent_id =
$(this).parent().parent().parent().parent().parent().find(".comment").find(".fa-
edit").data('id');
  $(".newRepply").remove();
  let repplyToUsername =
$(this).parent().parent().find(".topPartComment").find(".usernameComment").text();

  let userFaHtml = $("#leaveComm .userFaHolder").html();
  let topCommHtml = $("#leaveComm .topPartComment").html();

  let html = `

137


```

```

nt').length;

    let html = `<div class="newReply">
        <div class="userFaHolder">
            ${userFaHtml}
        </div>
        <div class="commentRightReply">
            <div class="topPartCommentReply">
                ${topCommHtml}
            </div>
            <textarea class="myCommentReply" placeholder="Leave a
reply..."></textarea>
        </div>
        <span class="fa fa-close"></span>
    </div>`;

    if(replies){
        console.log(replies);
        $(this).parent().parent().parent().parent().find(".repliesDiv").hide();
        let firstReplyObj =
$(this).parent().parent().parent().parent().find(".replyComment").first();
        $(this).parent().parent().parent().parent().find(".replyComment").hide();
        $(html).insertBefore(firstReplyObj);

$(this).parent().parent().parent().parent().find(".repliesDiv").slideDown();
    }else{

$(this).parent().parent().parent().parent().find(".repliesDiv").html(html);
    }

    if(!$ (this).parent().parent().parent().parent().find(".repliesDiv").is(":visible")){
    }

$(this).parent().parent().parent().parent().find(".repliesDiv").slideDown();
    }
    $(".newReply .fa-close").click(function(){
        console.log($(this).parent())
        $(this).parent().remove();
    });
    // $("body").on("click", ".newReply .fa-close", function(e){
    //     console.log(e.target.parentElement);
    //     console.log($(e.target).parents());
    //     $(this).parent().remove();
    // });
    $(".myCommentReply").keyup(function(event){
        if (event.keyCode === 13) {
            let text = $(this).val();
            let movie_id = $("#viewMoreCom").data("movid");
            postReply(parent_id, text, movie_id, $(this));
            let replyObj = $(this).parent().parent();
            console.log(replyObj);
            $(this).parent().html("Your reply was posted. <span class='fa fa-
check'></span>");
            replyObj.delay(2000).fadeOut(function(){
                replyObj.remove();
            });
        }
    });
}

```

```

function postReply(parent_id, text, movie_id, obj){
  console.log(obj.parent().parent().parent());
  let repliesDivObj = obj.parent().parent().parent();
  let numOfReplies = repliesDivObj.find(".replyComment").length;

  $.ajax({
    headers: {
      'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
    },
    url: "postReply",
    method: "post",
    data:{
      parent_id: parent_id,
      comment: text,
      idMovie: movie_id
    },
    success: function(data){
      let isNewlyPosted = 1;
      let html = printReplies([data.newComment], data.loggedUserId,
isNewlyPosted);
      if(numOfReplies){
        let firstReplyObj = repliesDivObj.find(".replyComment").first();
        $(html).insertBefore(firstReplyObj);
      }else{
        repliesDivObj.html(html);
      }
    },
    error: function(data){

    }
  })
}

var offsetComments = 0;

function viewMoreComments(e){
  e.preventDefault();
  let id= $(this).data('movid');
  offsetComments += 5;

  $.ajax({
    url: "viewMoreComments/" + id + "/" + offsetComments,
    method: "GET",
    dataType: 'json',
    success: function(data){
      if(data.comments.length){
        let len = $('.moreCommentsHolder').length;
        let html = printComments(data.comments,
data.loggedUserId);
        if(len){
          let obj = $("#peopleComments
.moreCommentsHolder").last();
          $("<div
class='moreCommentsHolder'></div>").insertAfter(obj);
          let newObj = $("#peopleComments
.moreCommentsHolder").last();
          newObj.hide();

```

```

        newObj.html(html);
        newObj.fadeIn();
    }else{
        let obj = $("#peopleComments
.commentReplyHolder").last();
        $("<div
class='moreCommentsHolder'></div>").insertAfter(obj);
        $(".moreCommentsHolder").hide();
        $(".moreCommentsHolder").html(html);
        $(".moreCommentsHolder").fadeIn();
    }
    $(".bottomPartComment .fa-edit").click(updateComment);
    $(".bottomPartComment .fa-trash").click(deleteComment);
    $(".bottomPartComment .reply").click(replyToComment);
    $(".bottomPartComment .fa-angle-
down").click(showReplies);
    }
    },
    error: function(error){
        console.log(error);
    }
});
}

function deleteComment(){
    let id = $(this).data('id');
    $(this).parent().parent().parent().parent().remove();//bilo je hide
    console.log("fff");
    $.ajax({
        url: "deleteComment/" + id,
        method: "GET",
        success: function(data){
            let stringNumComm = $("#commentsNumber").text();
            let numberComments = parseInt(stringNumComm);
            if(numberComments){
                numberComments--;
            }
            $("#commentsNumber").html(numberComments);
        },
        error: function(error){
            console.log(error);
        }
    });
}

function updateComment(){
    $(this).css("display", "none");
    let commentRightObj =
$(this).parent().parent().parent().find(".commentRight");
    let comment =
$(this).parent().parent().find(".commentTextDiv").find(".commText").text();
    //console.log(comment);
    let ta_html = `<textarea class='updateCommArea'>${comment}</textarea>`;

    let commentId = $(this).data('id');

    $(this).parent().parent().parent().find(".commentTextDiv").html(ta_html);
    $("<span class='fa fa-close'></span>").insertAfter(commentRightObj);

```

```

$("body").on("keyup", "#comments .updateCommArea", function(event){
    if (event.keyCode === 13) {
        let modifiedComm = $(this).val();
        let updateCommAreaObj = $(this);

        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url: "updateComment",
            method: "post",
            data:{
                newComment: modifiedComm,
                commentId: commentId
            },
            success: function(data){
                let commPostedObj =
updateCommAreaObj.parent().parent().parent().find('.commPosted');
                let editedObj = commPostedObj.parent().find(".editedComm");

                if(editedObj.length == 0){
                    $("<span
class=\"editedComm\">Edited</span>").insertAfter(commPostedObj);
                }
                updateCommAreaObj.parent().parent().parent().find('.fa-
close').css("display","none");
                updateCommAreaObj.parent().parent().parent().find('.fa-
edit').css("display", "initial");
                updateCommAreaObj.parent().html(`<p
class="commText">${modifiedComm}</p>`);
            },
            errors: function(err){

            }

        })
    }
});
}

function postComment(){
    let comment = $("#myComment").val();
    let idMovie = $("#postCommentForm .fa-pencil").data('movid');
    let _token = $("input[name='_token']").val();
    if(comment.trim() !== ""){
        $.ajax({
            url: "postComment",
            method: "post",
            data:{
                _token: _token,
                comment: comment,
                idMovie: idMovie
            },
            success: function(data){
                //console.log("uspeh");
                let numOfComments = $(".commentReplyHolder").length;
                console.log(numOfComments);
                if(numOfComments >= 5){

```

```

        console.log($(".commentReplyHolder").eq(5));
        $(".commentReplyHolder").eq(4).remove();
    }
    $("#myComment").val("");
    addNewlyPostedComment(data.newComment, data.loggedUserId);
},
error: function(error){
    //console.log("neuspeh");
}
})
}
}

function addNewlyPostedComment(comment, loggedUserId){
    let html = "";
    let stringNumComm = $("#commentsNumber").text();
    let numberComments = parseInt(stringNumComm);
    numberComments++;
    $("#commentsNumber").html(numberComments);

    html = printComments([comment], loggedUserId);
    // naravno da je korisnik koji je postavio kom ulogovan, medjutim bitno je da
    se prosledi i ovaj parametar
    // zbog ispisa ajaxom klikom na view more kada se takodje poziva funkcija
    printComments()

    let obj = $("#peopleComments .commentReplyHolder").first();
    console.log(obj);
    if(obj.length){
        $(html).insertBefore(obj);
    }else{
        // $("#peopleComments").append(html);
        $(html).insertAfter("#postCommentForm");
    }
    $(".bottomPartComment .fa-edit").click(updateComment);
    $(".bottomPartComment .fa-trash").click(deleteComment);
    $(".bottomPartComment .reply").click(replyToComment);
    $(".bottomPartComment .fa-angle-down").click(showReplies);
    $(".bottomPartComment .fa-angle-up").click(hideReplies);
}

function printComments(comments, loggedUserId){
    let html = ``;
    for(let comment of comments){
        html += `<div class="commentReplyHolder"><div class="comment">
            <div class="userFaHolder">
                <span class="fa fa-user`;
        if (comment.gender_id === 1){
            html += ` femaleUser`;
        }else{
            html += ` maleUser`;
        }
        html += `"></span>
            <p class="memberSince">Member since</p>`;
        let dateObj = new Date(comment.registered_at);
        let dateArray = formatDate(dateObj);
        html += `<p
class="memberDate">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}<
/p>

```

```

    </div>
    <div class="commentRight">
    <div class="topPartComment">
        <span class="usernameComment">${comment.username}</span>`;
        dateObj = new Date(comment.posted_at);
        dateArray = formatDate(dateObj);

        html += `<span
class="commPosted">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}.
${dateArray["hours"]}.${dateArray["minutes"]}</span>
    </div>
    <div class="commentTextDiv">
        <div class="commText">${comment.text}</div>
    </div>
    <div class="bottomPartComment">
        <span class="fa fa-angle-down showReplies angle" data-
id="${comment.id}"></span>
        <span class="fa fa-angle-up hideReplies angle" data-
id="${comment.id}"></span>`;
        if(loggedUserId){
            html += `<span class="repply" data-id="${comment.id}">Repply</span>`;
            if(loggedUserId === comment.user_id){
                html += `<span class="fa fa-edit" data-id="${comment.id}" data-
movid="${comment.idMovie}"></span>
                    <span class="fa fa-trash" data-id="${comment.id}"
data-movid="${comment.idMovie}"></span>`;
            }
        }
        html += `</div>
    </div>
    </div>
    <div class="repliesDiv"></div>
    </div>`;
    //pred poslednji div <span class='fa fa-close'></span>
}
return html;
}

var rbMoviesSortChecked;

function filterMovies(){
    var indexDate = $("#dateFilter").prop("selectedIndex");
    let dateValue = document.getElementById("dateFilter")[indexDate].value;
    // console.log(dateValue);
    var indexTechnology = $("#technologyFilter").prop("selectedIndex");
    let technValue =
document.getElementById("technologyFilter")[indexTechnology].value;
    var indexGenre = $("#genreFilter").prop("selectedIndex");
    let genreValue = document.getElementById("genreFilter")[indexGenre].value;
    let searched = $("#extendSearch").val();
    let rb = document.getElementsByName("sortMovies");

    for(let r of rb){
        if(r.checked){
            if(r.value !== rbMoviesSortChecked){
                rbMoviesSortChecked = r.value;
            }
        }
    }
}

```

```

$.ajax({
    url:"filterMovies",
    method: "get",
    data:{
        date: dateValue,
        technology: technValue,
        genre: genreValue,
        searched: searched.trim(),
        sort: rbMoviesSortChecked
    },
    dataType: "json",
    success: function(data){
        if(!data.movies.length){
            $("#moviesContainer").html("<p id='noResult'>No result found.
:(</p>");
        }else{
            printMovies(data.movies.slice(0,4));
            printPagination(data.movies, 4)
        }

        if($("#deleteFiltersDiv").hasClass("notVisibleFilterDiv")){
            $("#deleteFiltersDiv").removeClass("notVisibleFilterDiv");
            $("#deleteFiltersDiv").hide();
        }else{
            $("#deleteFiltersDiv").show();
        }

    },
    error:function(xhr, textStatus, errorThrown){
        //console.log(xhr.responseJSON.errors);
    }
});
console.log(dateValue + " " + technValue + " " + genreValue + " " + searched +
" " + rbMoviesSortChecked);
}

function printPagination(movies, limit){
    var length = movies.length;
    var i;
    var paginationHtml = "";
    var numOfPagLinks = Math.ceil(length/limit);
    for(i = 0; i < numOfPagLinks; i++){
        paginationHtml += `<li>
<a href="#" class="filteredPagination" data-limit="${i}"> ${i+1}</a>
</li>`;
    }
    $('#paginationDiv #pagination').html(paginationHtml);

document.getElementsByClassName('filteredPagination')[0].classList.add('activePagi
nation');
    $('#paginationDiv #pagination .filteredPagination').click(function(e){
        e.preventDefault();
        $('.filteredPagination').removeClass('activePagination');
        $(this).addClass('activePagination');
        var offset = $(this).data('limit');
        clickedPaginationFiltered(movies, offset);
    });
}

```



```

function clickedPaginationFiltered(movies, offset){
    var limit = 4;
    var printStart = limit * offset + 1;
    var printEnd = printStart + (limit - 1);
    console.log(printStart);
    console.log(printEnd);
    var i = 0;
    let html = '';
    let arrayToPrint = [];
    for(movie of movies){
        i++;
        if((i>=printStart) && (i<=printEnd)){
            arrayToPrint.push(movie);
        }
    }
    printMovies(arrayToPrint);
    $("html, body").animate({ scrollTop: $('#movies').position().top }, 1000);
}

function printMovies(movies){
    let html = "";
    for(let movie of movies){
        html += `<div class="movie">
                <div class="picGeneralMovieInfo">
                    <div class="moviePic">
                        <a href="${global_url}movie/${movie.idMovie}">`;
        if(movie.path.indexOf("cloudinary") != -1){
            html += ``;
        }else{
            html += ``;
        }

        html += `
                </a>
            </div>
            <div class="generalMovieInfo">
                <a href="${global_url}movie/${movie.idMovie}">
                    <h3> ${movie.movieName}<span class="ageLimit">
${movie.age_limit}</span></h3>
                </a>
                <span class="movieGrade">
                    <p class="numberMovieGrade">
${movie.grade}</p>
                <ul>`;
        html += writeStars(movie);
        html+= `</ul></span>
                <p class="movieDesc"> ${movie.description }</p>`;

        let i=0;
        for(let genre of movie.genres){
            i++;
            html += `<p>${genre.genre}`;
            if(movie.genres.length != i){
                html += `, `;
            }
        }
        html += `| <span class="fa fa-clock-o"></span> ${movie.running_time} min
            </p>
            </div>`;
    }
}

```

```

        <p class="movieDate">In cinemas from:`;
        let dateObj = new Date(movie.in_cinemas_from);
        let dateArray = formatDate(dateObj);

        html +=
`<span>${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}</span>
        </p>
        <a href="${global_url}movie/${movie.idMovie}"
class="detailsLink">View details</a>
        </div>
    </div>
    <div class="movieProjections">`;

    html += writeMovieProjections(movie.projections);

    html += `<div class="cleaner"></div>
    </div>
</div>
<div class="decorationalMovieDiv"></div>`;
    }
    $("#moviesContainer").html(html);
}

function writeMovieProjections(projections){
    let user = $("#hiddenUserExists").val();
    let html = "";
    for(let p of projections){
        console.log(p + " projekcija");
        console.log(p.reservation_available);
        console.log(p.projectionAlreadyStartedEnded);
        let toDate = new Date(p.starts_at);
        let hour = toDate.getHours();
        let minute = toDate.getMinutes();
        if(hour < 10)
        {
            hour = '0' + hour;
        }
        if(minute < 10){
            minute = '0' + minute;
        }
        html += `<div class="movieProjection`;
        if((!p.reservation_available) || p.projectionAlreadyStartedEnded)
        {
            html += " disabledReservation";
        }
        html += `>`;
        if(user != '0') {
            if(p.reservation_available && (!p.projectionAlreadyStartedEnded)) {
                html += ` <a href="${global_url}continueReservation/${p.id}" data-
id="${p.id}" class="projectionLink">`;
            }
        }
        html += `<p class="movieTime">
            ${hour}:${minute}</p>`;

        if((!p.reservation_available) || p.projectionAlreadyStartedEnded){
            html += `<span class="fa fa-ticket redTicket"></span>`;
        }else{
            html += `<span class="fa fa-ticket greenTicket"></span>`;
        }
    }
}

```

```

    }
    html += ` </p>
        <p class="movieHall">${p.name}</p>
        <p class="movieTechn">${p.technologyName}</p>`;
    if(user !== '0') {
        if(p.reservation_available && (!p.projectionAlreadyStartedEnded)) {
            html += `</a>`;
        }
    }
    html += `</div>`;
}
return html;
}
}

var slideShow1;

function changeSliderMovie(e){
    clearInterval(slideShow);
    clearInterval(slideShow2);
    console.log(indexPic + " " + moviesArray[indexPic].movieName);
    e.preventDefault();

    let idAttr = $(this).attr("id");
    console.log(idAttr);
    if(idAttr === "leftArrow"){
        if(!(indexPic === 0)){
            indexPic--;
        }else{
            indexPic = moviesArray.length - 1;
        }
    }else{
        if(indexPic >= moviesArray.length - 1){
            indexPic = 0;
        }else{
            indexPic++;
        }
    }

    console.log(indexPic);
    console.log(indexPic + " " + moviesArray[indexPic].movieName);
    printSliderInfo();
    clearInterval(slideShow1);
    slideShow1 = setInterval('slider()',4000);
}

function showSubmenu(e){
    let hrefAttr = $(this).attr("href");

    if(hrefAttr === "#"){
        let clickedOnSame = $(this).parent().find(".menuChildrenList:visible");

        if(clickedOnSame.length){
            $(this).parent().find(".menuChildrenList").removeClass("visible");
            $(this).parent().find(".menuChildrenList").addClass("notVisible");
            $(this).parent().removeClass("no-border");
        }else{
            $("#menuList .menuChildrenList").removeClass("visible");

```

```

        $("#menuList .menuChildrenList").addClass("notVisible");
        $("#menuList .menuChildrenList").removeClass("no-border");
        $(this).parent().find(".menuChildrenList").addClass("visible");
        $(this).parent().find(".menuChildrenList").removeClass("notVisible");
        $(this).parent().addClass("no-border");
    }
}

function extendSearch(){
    $("#movies .searchicon").css("margin-left","0px");
    $("#movies #extendSearch").removeClass("extend");
    $("#movies #extendSearch").addClass("extended");
    $("#movies #extendSearch").focus();
}

function showX(){
    $("#searchCont .fa-close").removeClass("notVisible");
    $("#searchCont .fa-close").addClass("visible");
}

function emptySearchInput(){
    $("#movies #extendSearch").val("");
    $("#searchCont .fa-close").removeClass("visible");
    $("#searchCont .fa-close").addClass("notVisible");
    $("#movies #extendSearch").focus();
    // filterMovies();
}

function deleteFilters(){
    $("#extendSearch").val("");
    $('.ddlFilter').find('option').each( function() {
        var $this = $(this);

        $this.removeAttr('selected');

    });

    document.getElementById("technologyFilter").children[0].setAttribute("selected","selected");

    document.getElementById("genreFilter").children[0].setAttribute("selected","selected");

    document.getElementById("dateFilter").children[0].setAttribute("selected","selected");
    $('#descOrder').prop('checked', false);
    $('#ascOrder').prop('checked', true);
    $("#deleteFiltersDiv").addClass("notVisibleFilterDiv")
    filterMovies();
}

function movieSlider(){
    $.ajax({
        url:"getSliderData",
        method: "get",
        dataType: "json",
        success: function(data){
            moviesArray = data.movies;

```

```

        slider();
    },
    error:function(xhr, textStatus, errorThrown){
        //console.log(xhr.responseJSON.errors);
    }
});
}

var indexPic = -1;
var moviesArray = []; //dodati = 0 da ne izlazi greska u konzoli !!!
function slider(){
    indexPic++;
    if(indexPic >= moviesArray.length){
        indexPic = 0;
    }

    printSliderInfo();
}

if($("#slider").length){
    var slideShow = setInterval('slider()',4000);
}

function printSliderInfo(){
    var pic= document.getElementById('sliderPicture');
    if(moviesArray[indexPic]){
        pic.src = urlPics + "/" + moviesArray[indexPic].picName;
        writeSliderActors(moviesArray[indexPic].actors);
        writeCenterSliderInfo(moviesArray[indexPic]);
        writeRightSliderInfo(moviesArray[indexPic]);
        writeSliderCircles();
    }
}

function writeSliderActors(actors){
    let html = "<span id='with'>With</span>\n";
    for(let a of actors){
        html += "<p>"
        let nameArray = a.name.split(" ");
        for(var i = 0; i < nameArray.length - 1; i++){
            html += `${nameArray[i]} `;
        }
        html += `<span class=\"actorLastname\">${nameArray[i]}</span></p>`;
    }

    $("#sliderActors").html(html);
}

function writeCenterSliderInfo(movie){
    let dateObj = new Date(movie.in_cinemas_from);
    let day = dateObj.getDate();
    let month = dateObj.getMonth();
    let year = dateObj.getFullYear();
    if(day < 10)
    {
        day = '0' + day;
    }
    if(month < 10){
        month = '0' + month;
    }

```

```

    }

    let html = `

## ${movie.movieName}</h2>\n` + ` <span class="fa fa-star"></span></li>`; } if((firstDecimal !== 0) && (firstDecimal >= 3)){ if(firstDecimal <= 7){ i++; html += `- <span class="fa fa-star-half-o"></span></li>`; }else{ i++; html += `- <span class="fa fa-star"></span></li>`; } } for(i; i < 10; i++){ html += `- <span class="fa fa-star-o"></span></li>`; } return html; } function writeSliderCircles(){ let html = ""; for(let i = 0; i < moviesArray.length; i++){ if(i === indexPic){ html += `- <a href="#" data-id="\${i}"><span class="fa fa-


```

```

circle"></span></a></li>`
    }else{
        html += `- <a href="#" data-id="${i}"><span class="fa fa-circle-
o"></span></a></li>`;
    }
}
$("#sliderCircles").html(html);
$("#sliderCircles li a").click(clickedSliderCircle);
}

var slideShow2;

function clickedSliderCircle(e){
    e.preventDefault();
    let idAttr = $(this).data("id");
    let numOfCircle = parseInt(idAttr);
    console.log(numOfCircle);
    clearInterval(slideShow);
    clearInterval(slideShow1);
    clearInterval(slideShow2);

    indexPic = numOfCircle;
    printSliderInfo();
    slideShow2 = setInterval('slider()', 4000);
}

function checkingRegData(){
    var name = document.querySelector("#regName").value.trim();
    var pass = document.querySelector("#regPass").value.trim();
    var pass1 = document.querySelector("#regPass1").value.trim();
    var email = document.querySelector("#email").value.trim();
    var tel = document.querySelector("#tel").value.trim();
    var genderArray=document.getElementsByName('gender');
    if(document.querySelector('#chbMail').checked){
        var sendViaMail=1;
    }else{
        sendViaMail=null;
    }

    var nameError = document.querySelector("#nameError");
    var passError = document.querySelector("#passError");
    var pass1Error = document.querySelector("#regPass1Error");
    var emailError = document.querySelector("#mailError");
    var telError = document.querySelector("#telError");
    var genderError = document.querySelector("#genderError");
    var townError = document.querySelector("#townError");
    var _token = $("input[name='_token']").val();

    var reName=/^[A-Za-z]{6,15}[\d]{1,5}$/;
    var rePass=/[\d\w\@$\%^*~\.\_\!+\=\)\(\{\}\|\?\/\<\>]{6,13}/;
    var reEmail=/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,4})+$/;
    var reTel=/^06[\d]\-[\d]{3}\-[\d]{3,4}$/;

    var errors = 0;
    if(name==""){
        nameError.innerHTML="Username field is required!";
        errors++;
    }else if(!reName.test(name)){

```

```

        nameError.innerHTML="Username must contain at least 6 characters and at
least 1 number.";
        errors++;
    }else{
        nameError.innerHTML="";
    }

    if(pass==""){
        passError.innerHTML="Password field is required.";
        errors++;
    }else if(!rePass.test(pass)){
        passError.innerHTML="Password must contain at least 6 characters..";
        errors++;
    }else{
        passError.innerHTML="";
    }

    if(pass1!=pass){
        errors++;
        pass1Error.innerHTML="You must type in your password again.";
    }else if(pass1==""){
        pass1Error.innerHTML="You must type in your password again.";
        errors++;
    }else{
        pass1Error.innerHTML="";
    }

    if(email==""){
        emailError.innerHTML="Email field is required!";
        errors++;
    }else if(!reEmail.test(email)){
        emailError.innerHTML="Email is not in valid format.";
        errors++;
    }else{
        emailError.innerHTML="";
    }

    if(tel==""){

    }else if(!reTel.test(tel)){
        telError.innerHTML="Tel needs to be in this format 06*-***-*****.";
        errors++;
    }else{
        telError.innerHTML="";
    }

    var selectedGender="";
    for(var i=0;i<genderArray.length;i++){
        if(genderArray[i].checked){
            selectedGender=genderArray[i].value;
            break;
        }
    }
    if(selectedGender==""){
        genderError.innerHTML="Choose gender.";
        errors++;
    }else{
        genderError.innerHTML="";
    }

```



```

console.log(errors);
if(errors === 0){
    $.ajax({
        url: global_url + "registration",
        method: "post",
        data: {
            _token: _token,
            username: name,
            pass: pass,
            pass1: pass1,
            email: email,
            tel: tel,
            selectedGender: selectedGender,
            sendViaMail: sendViaMail
        },
        success: function(data){
            successReg();
        },
        error: function(xhr, textStatus, errorThrown){
            //console.log(xhr.responseJSON.errors);
            //console.log(textStatus);
            //console.log(errorThrown);
            var error="An error occurred.";
            switch(xhr.status){
                case 404:
                    error="Page not found.";
                    break;
                case 422:
                    error="Your data is not in valid format.";
                    break;
                case 500:
                    error="There was an error.Try again.";
                    break;
                case 409:
                    error="<span class='fa fa-exclamation-triangle' aria-
hidden='true'> </span> Someone with same username or email already exists.";
                    break;

            }
            console.log(error);
            let errorsResponse = xhr.responseJSON;
            let errorsResponseText = "";
            $.each(errorsResponse.errors,function (k,v) {
                errorsResponseText += v + "<br/>";
            });
            let allErrors = errorsResponseText + "<br/>" + error;
            document.querySelector("#registrationForm
.success").innerHTML=allErrors;
            document.querySelector("#registrationForm
.success").classList.add('errorRegistration');
        }
    });
}else{
    document.querySelector("#registrationForm .success").innerHTML="";
}
function successReg(){
    document.querySelector("#registrationForm .success").innerHTML='<span
class="fa fa-check"></span> You are successfully registrated.';
    if($("#registrationForm .success").hasClass('errorRegistration'))

```

```

        document.querySelector("#registrationForm
.success").classList.remove('errorRegistration');
        disappearReg=setInterval('disappear()',3000);
    }
}

function disappear(){
    $("#blurBackground").fadeOut();
    clearInterval(disappearReg);
}

// ***** ADMIN *****

function sortActivity(){
    let sort =
document.getElementById('activityList').options[document.getElementById("activityL
ist").selectedIndex].value;
    $.ajax({
        url: "sortActivity",
        method: "GET",
        data: {
            sort: sort
        },
        success: function (data) {
            console.log(data);
            printActivities(data.activities);
        },
        error: function (data) {
            console.log(data);
        }
    });
}

function printActivities(activities){
    let html = "<tr><th>Activity</th><th>Date</th></tr>";
    for(let a of activities){
        html += `<tr>
                <td>${ a.text }</td>
                <td>${ a.activity_time }</td>
            </tr>`;
    }
    $("#activityTable").html(html);
}

function deleteAdminComment(){
    let id = $(this).data("id");
    let obj = $(this);
    $.ajax({
        url: "deleteComment/" + id,
        method: "get",
        success: function(data){
            obj.parent().parent().remove();
        },
        error: function(err){

        }

    });
}

```

```

function checkInsert(e){
    e.preventDefault();
    let text = $("#textLinkIns").val().trim();
    let path = $("#pathLinkIns").val().trim();
    let header = $("#headerLinkIns option:selected").val();
    let footer = $("#footerLinkIns option:selected").val();
    let parent = $("#parentLinkIns option:selected").val();
    let admin = 0;
    if($("#adminLinkIns").is(":checked")){
        admin = 1;
    }
    var _token = $("input[name='_token']").val();
    let errors = [];

    if(text == ""){
        errors.push("Text field is required!");
    }
    if(path == ""){
        errors.push("Path field is required!");
    }

    if(errors.length){
        let errHtml = "";
        for(let e of errors){
            errHtml += e + " ";
        }
        $("#insertLinkForm .error").html(errHtml);
        $("#insertLinkForm .success").html("");
    }else{
        $("#insertLinkForm .error").html("");
        $.ajax({
            url: "insertLink",
            method: "POST",
            data: {
                _token: _token,
                text: text,
                path: path,
                header: header,
                footer: footer,
                parent: parent,
                admin: admin
            },
            dataType: "json",
            success: function(data){
                $("#insertLinkForm .error").html("");
                $("#insertLinkForm .success").html("Link was successfully
inserted.");
                printLinks(data.links);
            },
            error: function(err){
                $("#insertLinkForm .success").html("");
                $("#insertLinkForm .error").html("There was an error.");
            }
        })
    }
}

```

```

function updateLink(e){
    e.preventDefault();
    let id = $("#linkIdHidden").val();
    let text = $("#textLinkUpd").val().trim();
    let path = $("#pathLinkUpd").val().trim();
    let header = $("#headerLinkUpd option:selected").val();
    let footer = $("#footerLinkUpd option:selected").val();
    let parent = $("#parentLinkUpd option:selected").val();
    let admin = 0;
    if($("#adminLinkUpd").is(":checked")){
        admin = 1;
    }
    var _token = $("input[name='_token']").val();
    let errors = [];

    if(text == ""){
        errors.push("Text field is required!");
    }
    if(path == ""){
        errors.push("Path field is required!");
    }

    if(errors.length){
        let errHtml = "";
        for(let e of errors){
            errHtml += e + " ";
        }
        $("#updateLinkForm .error").html(errHtml);
        $("#updateLinkForm .success").html("");
    }else{
        $("#updateLinkForm .error").html("");
        $.ajax({
            url: "updateLink",
            method: "POST",
            data: {
                _token: _token,
                id: id,
                text: text,
                path: path,
                header: header,
                footer: footer,
                parent: parent,
                admin: admin
            },
            success: function(data){
                $("#updateLinkForm .error").html("");
                $("#updateLinkForm .success").html("Link was successfully
updated.");
                getLinks();
            },
            error: function(err){
                $("#updateLinkForm .success").html("");
                $("#updateLinkForm .error").html("There was an error.");
            }
        })
    }
}

```

```

function getLinks(){
    $.ajax({
        url: "getLinks",
        method: "GET",
        success: function(data){
            printLinks(data.links);
        },
        error: function(err){

        }

    })
}

function printLinks(links){
    let html = `<tr>
        <th>Num.</th>
        <th>Text</th>
        <th>Path</th>
        <th>In header</th>
        <th>In footer</th>
        <th>Admin menu</th>
        <th></th>
    </tr>`;

    let i = 0;
    for(let l of links){
        i++;
        html += `<tr>
            <td>${i}</td>
            <td>${l.text}</td>
            <td>${l.href}</td>
            <td>`;

        if(l.in_header){
            html += `Yes.`;
        }else{
            html += `No.`;
        }

        html += `</td>
            <td>`;

        if(l.in_footer){
            html += `Yes.`;
        }else{
            html += `No.`;
        }

        html += `</td>
            <td>`;

        if(l.admin_link){
            html += `Yes.`;
        }else{
            html += `No.`;
        }

        html += `</td>
            <td>
                <div><span class="fa fa-edit" data-id="${l.menu_id}"></span></div>
                <div><a href="${global_url}deleteLink/${l.menu_id}"><span class="fa
fa-remove"></span></a></div>

```

```

        </td>
    </tr>`;
}
$("#linksTable").html(html);
$("#linksTable .fa-edit").click(editLinkClick);
}

function editLinkClick(){
    let id = $(this).data("id");
    $.ajax({
        url: "getLinkData/" + id,
        method: "get",
        dataType: "json",
        success: function(data){
            console.log(data.link);
            let link = data.link[0];
            $("#linkIdHidden").val(id);
            $("#textLinkUpd").val(link.text);
            $("#pathLinkUpd").val(link.href);
            $("#headerLinkUpd option[value='" + link.in_header +
            "'").prop('selected', true);
            $("#footerLinkUpd option[value='" + link.in_footer +
            "'").prop('selected', true);
            $("#parentLinkUpd option[value='" + link.idParent +
            "'").prop('selected', true);
            if(link.admin_link){
                $("#adminLinkUpd").prop("checked", true);
            }else{
                $("#adminLinkUpd").prop("checked", false);
            }
            $('#blurBgUpdateLink').fadeIn();
            $(document).bind('scroll');
            $('body').css({'overflow':'hidden'});

        },
        error: function(err){
        }
    })
}

function printUsers(users){
    let html = `<tr>
                <th>Num.</th>
                <th>Username</th>
                <th>Email</th>
                <th>Password</th>
                <th>Role</th>
                <th>Send news</th>
                <th>Gender</th>
                <th></th>
            </tr>`;
    for(let i = 0; i < users.length; i++){
        html += `<tr>
                <td>${i}</td>
                <td>${users[i].username}</td>
                <td>${users[i].email}</td>
                <td>${users[i].password}</td>
                <td>${users[i].roleName}</td>
                <td>`;
    }

```

```

        if(users[i].send_via_mail){
            html += `Send.`;
        }else{
            html += `Don't send.`;
        }

        html += `</td>`
        <td>`;
        if(users[i].genderName == 'f'){
            html += `Female.`;
        }else{
            html += `Male.`;
        }

        html += `</td>`
        <td>
            <div><span class="fa fa-edit" data-
id="${users[i].idUser}"></span></div>
            <div><a href="${global_url}deleteUser/${users[i].idUser}"><span
class="fa fa-remove"></span></a></div>
            </td>
        </tr>`;
    }
    $("#tableUsers").html(html);
}

function getUsers(){
    $.ajax({
        url: "getUsers",
        method: "GET",
        dataType: "json",
        success: function(data){
            printUsers(data.users);
        },
        error: function(err){
        }
    });
}

function checkUpdateUser(e){
    e.preventDefault();
    let id = $("#userIdHidden").val();
    let username = $("#usernameUpd").val();
    let password = $("#passwordUpd").val().trim();
    let role = $("#roleUserUpd option:selected").val();
    let email = $("#emailUserUpd").val();
    let gender = $("#genderUserUpd option:selected").val();
    let news = $("#newsUserUpd option:selected").val();
    var _token = $("input[name='_token']").val();
    let errors = [];

    var reName=/^[A-Za-z]{6,15}[\d]{1,5}$/;
    var rePass=/[\d\w\@\$%\^*\~\.\_\!\\+=\)\(\{\}\|\?\/\<\>]{6,13}/;
    var reEmail=/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,4})+$/;

    if(username==""){
        errors.push("Username field is required.")
    }else if(!reName.test(username)){
        errors.push("Username field is not valid.")
    }

```

```

    }

    if(password!="") {
        if (!rePass.test(password)) {
            errors.push("Password field is not valid.")
        }
    }
    console.log(errors);
    if(errors.length){
        let errHtml = ``;
        for(let e of errors){
            errHtml += e;
        }
        $("#updateUserForm .errors").html(errHtml)
    }else{
        $.ajax({
            url: "updateUser",
            method: "POST",
            data:{
                _token: _token,
                id: id,
                username: username,
                password: password,
                email: email,
                role: role,
                gender: gender,
                news: news
            },
            dataType: "json",
            success: function(data){
                $("#updateUserForm .error").html("");
                $("#updateUserForm .success").html("User update was successful.");
                getUsers();
            },
            error: function(err){
                $("#updateUserForm .success").html("");
                $("#updateUserForm .error").html("There was an error.");
            }
        });
    }
}

function getUser(){
    let id = $(this).data("id");
    $.ajax({
        url: "getUser/" + id,
        method: "get",
        success: function(data){
            console.log(data);
            $("#userIdHidden").val(data.user[0].id);
            $("#usernameUpd").val(data.user[0].username);
            $("#roleUserUpd option[value='" + data.user[0].role_id +
            "']").prop('selected', true);
            $("#emailUserUpd").val(data.user[0].email);
            $("#genderUserUpd option[value='" + data.user[0].gender_id +
            "']").prop('selected', true);
            $("#newsUserUpd option[value='" + data.user[0].send_via_mail +
            "']").prop('selected', true);

```



```

        $('#blurBgUpdateUser').fadeIn();
        $(document).bind('scroll');
        $('body').css({'overflow':'hidden'});
    },
    error: function(err){

    }
});
}

function checkProjectionUpd(){
    let listIndex = $("#movieProjectionUpdate option:selected").index();
    let errors = [];

    if(!listIndex){
        errors.push("Choose movie. ");
    }
    let dateProj = $("#dateProjectionUpdate").val();
    if(dateProj == ""){
        errors.push(" Choose date. ");
    }
    let timeProj = $("#timeProjectionUpdate").val();
    let reTime = /^[\\d]{2}\\:[\\d]{2}$/;
    if(timeProj==""){
        errors.push(" Choose projection time. ");
    }else if(!reTime.test(timeProj)){
        errors.push(" Projection time is not valid ");
    }
    let technProjIndex = $("#technProjectionUpdate option:selected").index();
    //console.log(technProjIndex);
    if(!technProjIndex){
        errors.push(" Choose technology. ");
    }
    let hallProjIndex = $("#hallProjectionUpdate option:selected").index();
    if(!hallProjIndex){
        errors.push(" Choose hall. ");
    }
    console.log(errors);
    if(errors.length){
        let errHtml = ``;
        for(let e of errors){
            errHtml += e;
        }
        $("#updateProjForm .errors").html(errHtml)
        return false;
    }else{
        return true;
    }
}

function searchProjectionsByMovie(event){
    if (event.keyCode === 13) {
        event.preventDefault();
        let searched = $("#searchProjections").val().trim();
        $.ajax({
            url: "getProjections",
            method: "GET",
            data: {
                searched: searched
            }
        })
    }
}

```

```

    },
    success: function(data){
        // console.log(data.projections);
        printProjectionsAdmin(data.projections);
        $(".prInfo .fa-edit").click(editProjClick);
    },
    error: function(err){

    }
}
})
}
}

function printProjectionsAdmin(projections){
    let html = ``;
    for(let p of projections){
        let dateObj = new Date(p.starts_at);
        let dateArray = formatDate(dateObj);
        html += `<div class="projectionViewRow">
                <div class="prMovie prInfo">${p.name}</div>
                <div class="prDate
prInfo">${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}.
${dateArray["hours"]}:${dateArray["minutes"]}</div>
                <div class="prTechn prInfo">${p.technName}</div>
                <div class="prHall prInfo">${p.hallName}</div>
                <div class="prRes prInfo">`;

        if(p.reservation_available){
            html += `Available.`;
        }else{
            html += `Not available.`;
        }

        html += `</div>
                <div class="prUPdDel prInfo lastPr">
                    <span class="fa fa-edit" data-id="${p.id}"></span>
                    <a href="${global_url}deleteProjection/${p.id}"><span
class="fa fa-remove"></span></a>
                </div>
            </div>
            <div class="projAdmBorder"></div>`;
    }
    $("#projectionViews").html(html);
}

function editProjClick(){
    let id = $(this).data("id");
    $.ajax({
        url: "getProjectionData/" + id,
        method: "GET",
        dataType: "json",
        success: function(data){
            let projection = data.projection;
            console.log(projection);
            let html = `<option value="0">Choose technology..</option>`;
            console.log(data.technologies);
            for(let t of data.technologies){
                html += `<option value="${t.id}">${t.name}</option>`;
            }
            $("#technProjectionUpdate").html(html);
        }
    });
}

```

```

        $("#movieProjectionUpdate option[value='" + projection.idMovie +
        "']").prop('selected', true);
        $("#hallProjectionUpdate option[value='" + projection.idHall +
        "']").prop('selected', true);
        $("#technProjectionUpdate option[value='" + projection.idTechnology +
        "']").prop('selected', true);
        $("#resProjectionUpdate option[value='" +
        projection.reservation_available + "']").prop('selected', true);
        // let dateobj = new Date(projection.starts_at);
        // console.log(dateobj);
        let startsAt = projection.starts_at.split(" ");
        let dateArray = startsAt[0].split("-");
        let timeArray = startsAt[1].split(":");

        $("#dateProjectionUpdate").val(dateArray[0] + "-" + dateArray[1] + "-"
+ dateArray[2]);
        $("#timeProjectionUpdate").val(timeArray[0] + ":" + timeArray[1]);
        $("#idProjHidden").val(id);
        $("#blurUpdateProj").fadeIn();
        $('body').css({'overflow': 'hidden'});
        $(document).bind('scroll', function () {
            // window.scrollTo(0,0);
        });
    },
    error: function(err){

    }
})
}

function getMovieTechnologies(obj){
    let index = obj.find("option:selected")[0].index;
    let movieId = obj.find("option:selected")[0].value;
    if(index){
        $.ajax({
            url: "getMovieTechnologies/" + movieId,
            method: "get",
            dataType: "json",
            success: function(data){
                // console.log(data.technologies);
                let html = `<option value="0">Choose technology..</option>`;
                for(let t of data.technologies){
                    html += `<option value="${t.id}">${t.name}</option>`;
                }
                obj.parent().parent().find(".projectionTechnology").html(html);
            },
            error: function(err){

            }
        })
    }else{
        obj.parent().parent().find(".projectionTechnology").html("<option
value='0'>Choose technology..</option>");
    }
}

function checkProjectionsInsert(){
    let projections = $(".adminProjection").length;
    let errors = 0;
    for(let i = 0; i < projections; i++){

```

```

        let listIndex = $(".projectionMovie")[i].selectedIndex;
        let errorHtml = "";
        let error = "";
        if(!i){
            error =
$(".projectionMovie").eq(i).parent().parent().parent().children()[1];
        }else{
            error =
$(".projectionMovie").eq(i).parent().parent().parent().children()[2];
        }

        error.innerHTML = "";
        if(!listIndex){
            errors++;
            error.innerHTML = "Choose movie. ";
        }
        let dateProj = $(".projectionDate").eq(i).val();
        if(dateProj == ""){
            errors++;
            errorHtml = error.innerHTML;
            error.innerHTML = errorHtml + " Choose date.";
        }
        let timeProj = $(".projectionTime").eq(i).val();
        let reTime = /^[\\d]{2}\\:[\\d]{2}$/;
        if(timeProj==""){
            errors++;
            errorHtml = error.innerHTML;
            error.innerHTML = errorHtml + " Choose projection time. ";
        }else if(!reTime.test(timeProj)){
            errors++;
            errorHtml = error.innerHTML;
            error.innerHTML = errorHtml + " Projection time is not valid.";
        }
        let technProjIndex = $(".projectionTechnology")[i].selectedIndex;
        if(!technProjIndex){
            errors++;
            errorHtml = error.innerHTML;
            error.innerHTML = errorHtml + "Choose technology.";
        }
        let hallProjIndex = $(".projectionHall")[i].selectedIndex;
        if(!hallProjIndex){
            errors++;
            errorHtml = error.innerHTML;
            error.innerHTML = errorHtml + "Choose hall.";
        }
        if(errors){
            return false;
        }else{
            return true;
        }
    }
}

function printMoreProjections(){
    let num = $(this).val();
    let counted = $("#projectionsForm .adminProjection").length;
    let numToPrint = num - counted;
    let optionsFirstDdlMovie = $("#projectionsForm
.adminProjection").first().find(".projectionMovie").html();

```

```

    let optionsFirstDdlHall = $("#projectionsForm
.adminProjection").first().find(".projectionHall").html();
    // console.log(optionsFirstDdlMovie);
    // console.log(numToPrint);
    let html = ``;
    if(numToPrint > 0){
        for(let i = numToPrint; i <= numToPrint; i++){
            html += `<div class="adminProjection">
                <div class="projectionX"><span class="fa fa-
close"></span></div>
                <div class="projectionRow">
                    <div class="projectionMovieLabel projectionLabel">
                        <label class="projectionLab">Movie:</label>
                        <select name="projectionMovie[]"
class="projectionMovie inputAdminClass">
                            ${optionsFirstDdlMovie}
                        </select>
                    </div>
                    <div class="projectionDateLabel projectionLabel">
                        <label class="projectionLab">Projection date:</label>
                        <input type="date" name="projectionDate[]"
class="projectionDate inputAdminClass"/>
                    </div>
                    <div class="projectionTimeLabel projectionLabel">
                        <label class="projectionLab">Projection time:</label>
                        <input type="text" name="projectionTime[]"
class="projectionTime inputAdminClass" placeholder="Projection time.."/>
                    </div>
                    <div class="projectionTechnologyLabel projectionLabel">
                        <label class="projectionLab">Technology:</label>
                        <select name="projectionTechnology[]"
class="projectionTechnology inputAdminClass">
                            <option value="0">Choose technology..</option>
                        </select>
                    </div>
                    <div class="projectionHallLabel projectionLabel">
                        <label class="projectionLab">Hall:</label>
                        <select name="projectionHall[]"
class="projectionHall inputAdminClass">
                            ${optionsFirstDdlHall}
                        </select>
                    </div>
                    <div class="projectionResAvailableLabel projectionLabel">
                        <label class="projectionLab">Reservation?</label>
                        <select name="projectionResAvailable[]"
class="projectionResAvailable inputAdminClass">
                            <option value="0">Not available.</option>
                            <option value="1"
selected="true">Available.</option>
                        </select>
                    </div>
                </div>
                <p class="error"></p>
            </div>`;
        }
        let lastObj = $("#projectionsForm .adminProjection").last();
        $(html).insertAfter(lastObj);
    }else{
        $("#projectionsForm .adminProjection").last().remove();
    }

```

```

    }
}

function checkMovieInsert(){
    let name = $("#nameMovIns").val().trim();
    let running_time = $("#timeMovIns").val().trim();
    let desc = $("#taShortDescIns").val().trim();
    let detailed_desc = $("#taDetailedDescIns").val().trim();
    let distributer = $("#distributerMovIns option:selected").index();
    let director = $("#directorMovIns option:selected").index();
    let country = $("#countryMovIns option:selected").index();
    let age_limit = $("#ageMovIns").val();
    let actors = $("#actorsListMovIns").find(".movUpdActGenres").length;
    let genres = $("#genresListMovIns").find(".movUpdActGenres").length;
    let date = $("#dateMovIns").val();

    let reTime = /^[\\d]{2,3}$/;

    var errors = [];

    if(date == ""){
        errors.push("In cinemas from field is required!");
    }
    if(name==""){
        errors.push("Movie name field is required!");
    }

    if(running_time==""){
        errors.push("Running time field is required!");
    }else if(!reTime.test(running_time)){
        errors.push("Running time is not valid.");
    }

    if(desc === ""){
        errors.push("Description cant't be empty.");
    }

    if(detailed_desc === ""){
        errors.push("Description detailed cant't be empty.");
    }

    if(!distributer){
        errors.push("Choose distributer.");
    }
    if(!director){
        errors.push("Choose director.");
    }
    if(!country){
        errors.push("Choose country.");
    }
    if(!actors){
        errors.push("Choose actors.");
    }
    if(!genres){
        errors.push("Choose genres.");
    }

    if(age_limit==""){
        errors.push("Age limit field is required!");
    }
}

```

```

    }else if(!reTime.test(age_limit)){
        errors.push("Age limit is not valid.");
    }

    if(errors.length){
        let errHtml = "";
        for(let e of errors){
            errHtml += e + " ";
        }
        $("#insertMovieForm .errors").html(errHtml);
        return false;
    }else{
        return true;
    }
}

function addMovieActorInsert(){
    let index = $(this).prop('selectedIndex');
    if(index){
        let id = $("#actorsMovIns option:selected").val();
        let name = $("#actorsMovIns option:selected").text();
        let last = $("#actorsListMovIns").find(".movUpdActGenres").last();
        let write = `<div class="movUpdActGenres"><a href="#" data-
id="\${id}">\${name}</a><span class="fa fa-close"></span>
        <input type="hidden" class="actorInsertHidden"
name="actorInsertHidden[]" value="\${id}"/>
        </div>`;
        if(last.length){
            $(write).insertAfter(last);
        }else{
            write += `<div class='cleaner'></div>`;
            $("#actorsListMovIns").append(write);
        }
    }
}

function addMovieGenreInsert(){
    let index = $(this).prop('selectedIndex');
    if(index){
        let id = $("#genresMovIns option:selected").val();
        let name = $("#genresMovIns option:selected").text();
        let last = $("#genresListMovIns").find(".movUpdActGenres").last();
        let write = `<div class="movUpdActGenres"><a href="#" data-
id="\${id}">\${name}</a><span class="fa fa-close"></span>
        <input type="hidden" class="genreInsertHidden"
name="genreInsertHidden[]" value="\${id}"/>
        </div>`;
        if(last.length){
            $(write).insertAfter(last);
        }else{
            write += `<div class='cleaner'></div>`;
            $("#genresListMovIns").append(write);
        }
    }
}

function checkMovieUpdate(){
    let name = $("#nameMovUpd").val().trim();

```

```

let running_time = $("#timeMovUpd").val().trim();
let desc = $("#taShortDesc").val().trim();
let detailed_desc = $("#taDetailedDesc").val().trim();
let distributer = $("#distributerMovUpd option:selected").index();
let director = $("#directorMovUpd option:selected").index();
let country = $("#countryMovUpd option:selected").index();
let age_limit = $("#ageMovUpd").val();
let actors = $("#actorsListMovUpd").find(".movUpdActGenres").length;
let genres = $("#genresListMovUpd").find(".movUpdActGenres").length;
let date = $("#dateMovUpd").val();

let reTime = /^[0-9]{2,3}$/;

var errors = [];

if(date == ""){
    errors.push("In cinemas from field is required!");
}
if(name==""){
    errors.push("Movie name field is required!");
}

if(running_time==""){
    errors.push("Running time field is required!");
}else if(!reTime.test(running_time)){
    errors.push("Running time is not valid.");
}

if(desc === ""){
    errors.push("Description cant't be empty.");
}

if(detailed_desc === ""){
    errors.push("Description detailed cant't be empty.");
}

if(!distributer){
    errors.push("Choose distributer.");
}
if(!director){
    errors.push("Choose director.");
}
if(!country){
    errors.push("Choose country.");
}
if(!actors){
    errors.push("Choose actors.");
}
if(!genres){
    errors.push("Choose genres.");
}

if(age_limit==""){
    errors.push("Age limit field is required!");
}else if(!reTime.test(age_limit)){
    errors.push("Age limit is not valid.");
}

if(errors.length){

```



```

        let errHtml = "";
        for(let e of errors){
            errHtml += e + " ";
        }
        $("#updateMovieForm .errors").html(errHtml);
        return false;
    }else{
        return true;
    }
}

function addMovieGenre(){//UPDATE MOVIE
    let index = $(this).prop('selectedIndex');
    if(index){
        let id = $("#genresMovUpd option:selected").val();
        let name = $("#genresMovUpd option:selected").text();
        let last = $("#genresListMovUpd").find(".movUpdActGenres").last();
        let write = `<div class="movUpdActGenres"><a href="#" data-
id="\${id}">\${name}</a><span class="fa fa-close"></span>
                        <input type="hidden" class="genreHidden" name="genreHidden[]"
value="\${id}"/>
                        </div>`;
        if(last.length){
            $(write).insertAfter(last);
        }else{
            write += `<div class='cleaner'></div>`;
            $("#genresListMovUpd").append(write);
        }
    }
}

function addMovieActor(){
    let index = $(this).prop('selectedIndex');
    if(index){
        let id = $("#actorsMovUpd option:selected").val();
        let name = $("#actorsMovUpd option:selected").text();
        let last = $("#actorsListMovUpd").find(".movUpdActGenres").last();
        // let count = $("#actorsListMovUpd").find(".movUpdActGenres").length;
        let write = `<div class="movUpdActGenres"><a href="#" data-
id="\${id}">\${name}</a><span class="fa fa-close"></span>
                        <input type="hidden" class="actorHidden" name="actorHidden[]"
value="\${id}"/>
                        </div>`;
        //id="actorHidden\${count}"
        if(last.length){
            $(write).insertAfter(last);
        }else{
            write += `<div class='cleaner'></div>`;
            $("#actorsListMovUpd").append(write);
        }
    }
}

function getMovieData(){
    let id = $(this).data("id");

    $.ajax({

```

```

url: "getMovieData",
method: "get",
data: {
  id: id
},
success: function (data) {
  console.log(data);
  let movie = data.movie;
  $('#blurBgUpdateMovie').fadeIn();
  $('body').css({'overflow': 'hidden'});
  $(document).bind('scroll', function () {
    window.scrollTo(0,0);
  });
  $("#updateMovieForm #movieIdHidden").val(id);
  $("#nameMovUpd").val(movie.movieName);
  $("#timeMovUpd").val(movie.running_time);
  $("#taShortDesc").val(movie.description);
  $("#taDetailedDesc").val(movie.description_detailed);
  $("#distributerMovUpd option[value='" + movie.distributer_id +
"]").prop('selected', true);
  $("#directorMovUpd option[value='" + movie.director_id +
"]").prop('selected', true);
  let technologies = movie.technologies;
  console.log(technologies);
  if(technologies.length > 1){
    $("#technologiesMovUpd option[value='" + 3 +
"]").prop('selected', true);
  }else{
    $("#technologiesMovUpd option[value='" + technologies[0].id +
"]").prop('selected', true);
  }
  $("#countryMovUpd option[value='" + movie.country_id +
"]").prop('selected', true);
  $("#dateMovUpd").val(movie.in_cinemas_from);
  $("#ageMovUpd").val(movie.age_limit);
  $("#activeSliderMovUpd option[value='" + movie.is_active_slider +
"]").prop('selected', true);
  let actors = "";
  for(let a of movie.actors){
    actors += `<div class="movUpdActGenres"><a href="#" data-
id="${a.id}">${a.name}</a><span class="fa fa-close"></span>
    <input type="hidden" class="actorHidden"
name="actorHidden[]" value="${a.id}"/>
    </div>`;
    //id="actorHidden${movie.actors.length}"
  }
  actors += `<div class="cleaner"></div>`;
  $("#actorsListMovUpd").html(actors);
  let genres = "";
  for(let g of movie.genres){
    genres += `<div class="movUpdActGenres"><a href="#" data-
id="${g.id}">${g.genre}</a><span class="fa fa-close"></span>
    <input type="hidden" class="genreHidden"
name="genreHidden[]" value="${g.id}"/>
    </div>`;
    //id="genderHidden${movie.genres.length}"
  }
  genres += `<div class="cleaner"></div>`;
  $("#genresListMovUpd").html(genres);

```

```

    },
    error: function (xhr, textStatus, errorThrown) {
    }
  });
}

function printAdminMoviePagination(movies){
  var length = movies.length;
  var i;
  var paginationHtml = "";
  var numOfPagLinks = Math.ceil(length/5);
  console.log(numOfPagLinks);
  for(i = 0; i < numOfPagLinks; i++){
    paginationHtml += `<li>
    <a href="#" class="paginationAdmMov" data-limit="${i}"> ${i+1}</a>
    </li>`;
  }
  $('#adminMovies #pagination').html(paginationHtml);

document.getElementsByClassName('paginationAdmMov')[0].classList.add('activePagina
tion');
}

function searchMoviesAdm(searched){
  if(searched == ""){
    let limit= 0;
    $.ajax({
      url: "admin/moviesPage/" + 0 + "/" + 0,
      method: "GET",
      dataType: 'json',
      success: function(data){
        printMoviesAdmin(data.movies);
        printAdminMoviePagination(data.allMovies);
      },
      error: function(error){
        console.log(error);
      }
    });
  }else{
    $.ajax({
      url: "admin/moviesPage/" + 0 + "/" + searched,
      method: "GET",
      dataType: 'json',
      success: function(data){
        // console.log(data.movies);
        printMoviesAdmin(data.movies);
        printAdminMoviePagination(data.movies);
      },
      error: function(error){
        console.log(error);
      }
    });
  }
}

function printMoviesAdmin(movies){
  // console.log(urlPics);
  let html = '';
  for(let m of movies){

```

```

    html += `<div class="movieAdmin">
        <div class="moviePicAdmin movieInfoAdmin">`;
    if(m.path.indexOf("cloudinary") != -1){
        html += ``;
    }else{
        html += ``;
    }

    html += `<p class="movieDateAdmin">Date aired: `;
    let dateObj = new Date(m.in_cinemas_from);
    let dateArray = formatDate(dateObj);

    html +=
`<span>${dateArray["day"]}.${dateArray["month"]}.${dateArray["year"]}</span>
    </p>
    <p>`;
    for(let i=0; i< m.genres.length; i++){
        html += `${m.genres[i].genre}`;
        if(m.genres.length-1 != i){
            html += ",";
        }
    }

    html += `| <span class="fa fa-clock-o"></span> ${m.running_time} min
    </p>
</div>
<div class="movieNameAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Movie name</p>
    ${m.movieName}
    <p class="headlineAdmMovie headlineAdmActors">Actors</p>
    <p class="actorsAdmMovie">`;
    for(let i=0; i<m.actors.length; i++){
        if(i === m.actors.length-1){
            html += `${m.actors[i].name}`;
        }else{
            html += `${m.actors[i].name}, `;
        }
    }

    html += `</p>
</div>
<div class="movieRatingAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Rating</p>
    <p class="numberMovieGrade">${m.grade}</p><ul>`;
    html+= writeStars(m);

    html += `</ul></div>
<div class="movieDirectorAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Director</p>
    ${m.dirName}
</div>
<div class="movieDescAdmin movieInfoAdmin">
    <p class="headlineAdmMovie">Description</p>
    <p class="admDescP">${m.description}</p>
</div>
<div class="movieDirectorAdmin movieInfoAdmin btnsAdminMovies">
    <div><input type="button" value="Update" class="btnUpdateMov" data-

```

```

id="${m.idMovie}"/></div>
    <div><a href="${global_url}deleteMovie/${m.idMovie}" data-
id="${m.idMovie}" class="btnDeleteMov">Delete</a></div>
    </div>`;
}
$("#adminMoviesHolder").html(html);
$(".btnUpdateMov").click(getMovieData);
}

```

4.7 CSS

4.7.1 Style.css

```

@font-face{
    font-family: 'SF';
    src: url('fonts/sf-new-republic/TrueType/SFNewRepublic.ttf');
}
@font-face{
    font-family: 'GO';
    src: url('fonts/goffik-outline/Goffik-O.ttf');
}
@font-face{
    font-family: 'PD';
    src: url('fonts/playfair-display/PlayfairDisplay-Regular.ttf');
}
*{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
body{
    font-family: 'Nunito Sans', sans-serif;
    color: #fff;
    background-color: #0f0d0c;
}
.cleaner{
    clear: both;
}
ul{
    list-style-type: none;
    display: flex;
}
a{
    color: #fff;
    text-decoration: none;
}
#blurBackground, #blurBackgroundReservation, #blurBackgroundPrices,
#blurBackgroundForgetPass, .blurBackgroundsAdmin{
    position: absolute;
    /*display: flex;*/
    z-index: 201;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.8);
    backdrop-filter: blur(2px);
}
#headerWrapper, #adminHeader{
    color: #fff;
}

```

```

    width: 100%;
    background-color: #1f1a19;
}
#header, #adminHeaderHolder{
    width: 80%;
    margin: 0px auto;
    padding: 13px 0px 7px 0px;
}
#logoMessg{
    width: 35%;
    display: flex;
    justify-content: space-between;
    float: left;
}
.logo{
    display: flex;
}
.logo h1, .logo h3{
    color: #e3002a;
    /*color: #C7181F;*/
    font-family: 'SF';
    font-style: italic;
    font-size: 35px;
    text-shadow: 2px 2px 4px #000;
}
.popcorn{
    background-image: url("../images/popcorn.svg");
    background-size: contain;
    width: 35px;
    height: 35px;
    margin-right: 3px;
    box-shadow: 0 4px 5px -3px #000;
}
#messageContainer{
    padding: 12px 0px 10px 0px;
}
#messageContainer .message{
    font-size: 13px;
    letter-spacing: 1px;
}
#hireList{
    right: 0px;
    top: 12%;
    position: fixed;
    width: 90px;
    display: block;
    z-index: 10;
}
#hireList li{
    background-color: rgba(0, 0, 0, 0.1);
    padding: 7px;
    border: 1px solid #fff;
    font-size: 13px;
    text-transform: uppercase;
    display: flex;
    justify-content: center;
}
#hireList li:nth-child(1){
    margin-bottom: 10px;
}

```

```

}
#hireList span {
  width: 30px;
  float: left;
  background-size: contain;
  margin: 2px 5px 0px 0px;
  background-repeat: no-repeat;
  background-position: center;
}
#hireList #ourPrices{
  background-image: url("../images/price-tag.svg");
}
#hireList #hireUs{
  background-image: url("../images/shake-hand.svg");
}
#logRegList #ourPrices{
  background-image: url("../images/price-tag.svg");
  width: 15px;
  height: 15px;
  float: left;
  background-size: contain;
  right: 0px;
  margin: 17px -8px 0px -10px;
  background-repeat: no-repeat;
  transform: rotate(-60deg);
  position: absolute;
  background-position: center;
}
#logReg{
  width: 20%;
  float: right;
  position: relative;
}
#logReg #logRegList{
  justify-content: space-between;
  margin-top: -5px;
}
#logRegList .fa-heart-o{
  font-size: 7px;
  padding: 5px 0px;
  color: #6b18d9 !important;
}
#logReg #logRegList a{
  font-size: 14px;
  color: #d6cfcf;
}
#logReg #logRegList .fa {
  margin-right: 3px;
  color: #fff;
}
#logReg #logRegList .fa-sign-in{
  color: #e3002a;
}
#logReg #logRegList #help{
  border-bottom: 1px dotted #444;
}
#logReg #logRegList #logLink, #logout{
  color: #e3002a !important;
}
}

```

```

#closeRegFormHolder, #closeResFormHolder, #closePricesHolder,
#closeForgetPassFormHolder, #closeUpdateMovieFormHolder,
#closeUpdateProjFormHolder{
    height: 70px;
}
#closeRegForm, #closeResForm, #closePrices,
#closeForgetPassForm, #closeUpdateMovieForm, #closeInsertMovieForm,
#closeUpdateProjForm, #closeUpdateUserForm, #closeUpdateLinkForm,
#closeInsertLinkForm{
    float: right;
    background-image: url("../images/close.svg");
    width: 25px;
    height: 25px;
    background-size: contain;
    background-repeat: no-repeat;
    margin: 25px 25px 0px 0px;
}
#closeRegForm:hover, #closeResForm, #closePrices,
#closeForgetPassForm, #closeUpdateMovieForm, #closeInsertMovieForm,
#closeUpdateProjForm, #closeUpdateUserForm:hover{
    cursor: pointer;
}
#registrationHolder, #reservationHolder, #pricesHolder, #forgetPassHolder,
#updateMovieHolder, #insertMovieHolder, #updateProjHolder, #updateUserHolder{
    margin: auto;
    background-color: #0a0a09;
}
#registrationHolder {
    width: 50%;
}

/* R A D I O */

/* The container */
.container {
    display: block;
    position: relative;
    padding-left: 35px;
    margin-bottom: 12px;
    cursor: pointer;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
}
#firstContainer{
    margin-top: 10px;
}
/* Hide the browser's default radio button */
.container input {
    position: absolute;
    opacity: 0;
    cursor: pointer;
}

/* Create a custom radio button */
.checkmark {
    position: absolute;
    top: 0;
    left: 0;
}

```



```

    height: 15px;
    width: 15px;
    background-color: #eee;
    border-radius: 50%;
}

/* On mouse-over, add a grey background color */
.container:hover input ~ .checkmark {
    background-color: #ccc;
}

/* When the radio button is checked, add a blue background */
.container input:checked ~ .checkmark {
    background-color: #fff;
}

/* Create the indicator (the dot/circle - hidden when not checked) */
.checkmark:after {
    content: "";
    position: absolute;
    display: none;
}

/* Show the indicator (dot/circle) when checked */
.container input:checked ~ .checkmark:after {
    display: block;
}

/* Style the indicator (dot/circle) */
.container .checkmark:after {
    top: 3px;
    left: 3px;
    /* right: 445px; */
    width: 10px;
    height: 10px;
    border-radius: 50%;
    /*background: #e3002a;*/
    /*background: #2d0c59;*/
    background-image: linear-gradient(#2d0c59, #6b18d9);
}

/* E N D   R A D I O*/

/* C H B*/

.containerChb {
    display: block;
    position: relative;
    padding-left: 35px;
    margin-bottom: 12px;
    cursor: pointer;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
}

/* Hide the browser's default checkbox */
.containerChb input {

```

```

    position: absolute;
    opacity: 0;
    cursor: pointer;
    height: 0;
    width: 0;
}

/* Create a custom checkbox */
.checkmarkChb {
    position: absolute;
    top: 0;
    left: 0;
    height: 15px;
    width: 15px;
    background-color: #eee;
}

/* On mouse-over, add a grey background color */
.containerChb:hover input ~ .checkmarkChb {
    background-color: #ccc;
}

/* When the checkbox is checked, add a blue background */
.containerChb input:checked ~ .checkmarkChb {
    /*background-color: #2196F3;*/
}

/* Create the checkmark/indicator (hidden when not checked) */
.checkmarkChb:after {
    content: "";
    position: absolute;
    display: none;
}

/* Show the checkmark when checked */
.containerChb input:checked ~ .checkmarkChb:after {
    display: block;
}

/* Style the checkmark/indicator */
.containerChb .checkmarkChb:after {
    left: 4px;
    top: -1px;
    width: 5px;
    height: 10px;
    border: solid #e3002a;
    border-width: 0 4px 4px 0;
    -webkit-transform: rotate(45deg);
    -ms-transform: rotate(45deg);
    transform: rotate(45deg);
}

/* E N D   C H B*/

.containerMoviesRadio{
    display: initial;
    padding-left: 20px;
    margin-left: 10px;
}

.checkmarkMovies{

```

```

    margin-top: 2px;
}
#registrationForm, #reservationForm{
    /*background-color: #0a0a09; !*#0f0d0c!**/
    z-index: 201;
    padding: 40px 70px;
    text-align: left;
    color: #757272;
}
#regHeadline, #resHeadline, #pricesHeadline, #forgetPassHeadline{
    background-image: url('../images/form-pattern.jpg');
}
#regHeadline #regHeadlineOpacity, #resHeadline #resHeadlineOpacity,
#pricesHeadline #pricesHeadlineOpacity, #forgetPassHeadlineOpacity{
    display: flex;
    justify-content: center;
    padding: 20px 0px;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.8);
}
#popcornRegForm{
    /*margin-left: 20px;*/
}
#regP, #resP, #pricesP, #forgetPassP{
    letter-spacing: 2px;
    text-align: center;
    font-size: 27px;
    font-family: 'SF';
    width: 70%;
}
.x{
    margin-left: 93%;
    font-size: 20px;
    padding-bottom: 10px;
    /*color: #eb34e1;*/
}
#registrationForm #male{
    margin-left: 10px;
}
#registrationForm table {
    margin: 0px auto;
    width: 100%;
}
#registrationForm #regName, #registrationForm #regPass, #registrationForm
#regPass1, #registrationForm #email, #registrationForm #tel, #registrationForm
#town, #registrationForm #address, #registrationForm #zip{
    width: 100%;
    height: 32px;
    font-size: 15px;
    padding-left: 20px;
    color: #757272;
    background-color: #0a0a09;
    border-width: 0px;
    border-bottom: 0.5px solid #757272;
    margin-bottom: 20px;
}
#registrationForm .marginTd{
    padding-left: 10px;
}

```

```

}
#registrationForm input {
    margin-bottom: 10px;
}
#registrationForm #btnRegistration {
    width: 100px;
    height: 35px;
    /*background-image: linear-gradient(#2d0c59, #6b18d9);*/
    background-color: #0a0a09;
    color: #fff;
    border: 0.5px solid #757272;
    border-radius: 5px;
    font-size: 15px;
    display: block;
    margin: 10px auto 0px auto;
}
#registrationForm #btnRegistration, .x:hover, #logForm #logIn, .fa-shopping-bag{
    cursor: pointer;
}
.error {
    color: #e3002a;
    font-size: 12px;
    padding-bottom: 5px;
}
.errorRegistration{
    color: #e3002a !important;
    margin-top: 10px;
}
.success{
    color:#6b18d9;
    font-weight:bold;
}
#closeLogForm{
    height: 25px;
}
#logFormX{
    float: right;
    background-image: url('../images/close.svg');
    width: 12px;
    height: 12px;
    background-size: contain;
    background-repeat: no-repeat;
    margin: 10px 12px 10px 0px;
}
#logFormX:hover{
    cursor: pointer;
}
#logFormDiv{
    position: absolute;
    z-index: 200;
    width: 180px;
    top: 25px;
    margin-top: 8px;
    background-image: linear-gradient(#0f0d0c, #1f1a19);
}
#logForm{
    padding: 0px 20px 10px 20px;
}
#logForm #logUsername,#logForm #logPass{

```

```

width: 100%;
height: 25px;
margin-bottom: 10px;
font-size: 14px;
/*border-radius: 5px;*/
background-color: transparent;
padding: 8px 0px 8px 7px;
border-width: 0px;
border-bottom: 0.5px solid #757272;
color: #757272;
}
#resetPass{
color: #757272;
font-size: 12px;
}
#logForm #logIn{
padding: 8px;
color: #fff;
border-radius: 5px;
border-width: 0px;
/*margin-bottom: 5px;*/
font-size: 14px;
background-image: linear-gradient(#2d0c59, #6b18d9);
display: block;
margin: 10px auto 10px auto;
}
#pricesHolder{
padding-bottom: 40px;
}
#pricesTable{
width: 80%;
margin: 40px auto 0px auto;
}
#pricesTable tr:nth-child(odd){
background-color: #141413;
}
#pricesTable td{
padding: 5px 0px 5px 10px;
}
#pricesTable th{
color: #e3002a;
padding: 7px 0px;
letter-spacing: 1px;
font-weight: bold;
font-family: 'SF';
}
#menuBtnDiv{
display: none;
}
#menu, #menuAdminDiv{
padding: 7px 0px;
background-color: #0f0d0c;
}
#menuList, #menuAdmin{
width: 80%;
margin: 0px auto;
justify-content: center;
}
#menuList li, #menuAdmin li{

```

```

    position: relative;
}
#menuList li a, #menuAdmin li a{
    padding-right: 25px;
}
#menuList > li > a:after{
    font-size: 10px;
    font-family: 'Nanum Gothic', sans-serif;
    content: '\25CF';
    color: #777;
    padding-left: 25px;
}
#menuList > li:last-child a:after, #menuAdmin > li:last-child a:after{
    content: '';
}
#menuList li a .fa-angle-down{
    margin: 3px 0px 0px 3px;
    font-size: 13px;
    color: #d6cfcb;
}
.visible{
    display: block;
}
.menuChildrenList{
    position: absolute;
    z-index: 100;
    background-color: #0f0d0c;
}
.notVisible{
    display: none;
}
.menuChildrenList li{
    white-space: nowrap;
    padding: 7px 0px 7px 7px;
    border-bottom: 1px solid #757272;;
}
#slider{
    /*background-image: url("../images/tenet3.jpg");*/
    height: 600px;
    background-size: cover;
    position: relative;
    box-shadow: 0 15px 15px #0f0d0c;
    background-position-x: center;
}
#slider #sliderPicture{
    /*max-width: 100%;*/
    width: 100%;
    height: 100%;
    object-fit: cover;
    /*position: absolute;*/
    /*left: 0;*/
    /*top: 0;*/
    /*width: 100%;*/
}
#slider #sliderInfoContainer{
    position: absolute;
    height: 100%;
    width: 100%;
    background-color: rgba(0, 0, 0, 0.8);
}

```

```

    top: 0;
}
#slider #sliderInfoContainer #sliderInfoWrapper{
    padding: 100px 0px;
    width: 100%;
    display: flex;
    justify-content: space-around;
    align-items: center;
    max-height: 556px;
}
#slider #sliderInfo{
    width: 80%;
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
#slider .fa-angle-right, #slider .fa-angle-left{
    font-size: 35px;
}
#slider #sliderActors{
    margin-top: -200px;
    font-size: 16px;
    color: #dbced9;
    line-height: 1.4em;
    letter-spacing: 1px;
    min-width: 230px;
    /*width: 15%;*/
}
#slider #sliderActors p{
    margin-bottom: 15px;
    font-weight: 600;
}
#slider #with{
    display: block;
    margin-left: 35px;
    padding-bottom: 15px;
}
#slider #sliderActors p:nth-child(1){
    margin-top: 7px;
}
#slider #sliderActors p:nth-child(2n){
    text-align: right;
}
#slider #sliderActors p:nth-child(1n){
    text-align: center;
}
#slider #sliderActors .actorPic{
    width: 40px;
    height: 40px;
    border-radius: 50%;
}
#slider #sliderActors .actorLastname {
    font-size: 17px;
    font-weight: 700;
    text-transform: uppercase;
    letter-spacing: 2px;
}
#slider #centerSliderInfo{

```

```

    text-align: center;
    width: 35%;
    max-height: 317px;
}
#slider #centerSliderInfo:before{
    content: url("../images/title-flourish-start.png");
    display: block;
    margin: auto;
    width: 50%; /* 200px*/
    border-bottom: 2px solid #ada8ad;
    text-align: center;
}
#slider #centerSliderInfo:after{
    padding-top: 5px;
    content: url("../images/title-flourish-end.png");
    display: block;
    margin: auto; /*350px*/
    width: 100%;
    border-top: 2px solid #ada8ad;
    text-align: center;
}
#centerSliderInfo h2{
    line-height: 1.2em;
    letter-spacing: 2px;
    font-size: 35px;
    margin-top: 30px;
}
#centerSliderInfo #director{
    font-size: 22px;
    letter-spacing: 1px;
    font-family: 'PD';
    /*font-family: 'Rochester', cursive;*/
}
#centerSliderInfo #inCinemasFrom{
    font-size: 14px;
    margin-top: 5px;
    color: #a0aec0;
}
#centerSliderInfo #inCinemasFrom .fa-calendar{
    margin-right: 5px;
}
#centerSliderInfo #bookSliderMovie{
    text-transform: uppercase;
    padding: 10px 15px;
    color: #fff;
    margin: 7px auto;
    /*background-color: #9c0085;*/
    border: 0px solid #eee;
    font-size: 13px;
    letter-spacing: 2px;
    margin: 15px 0px 30px 0px;
    /*background-image: linear-gradient(#4f0444, #b5049b);!#9c0085!*/
    background-image: linear-gradient(#2d0c59, #6b18d9);
    border-radius: 7px;
}
#centerSliderInfo #bookSliderMovie .fa-angle-right{
    margin-left: 7px;
    font-size: 14px;
}

```



```

#sliderGrades{
  /*width: 16%;*/
  font-size: 17px;
  margin-top: -150px;
}
#sliderGrades #sliderGrade{
  font-size: 45px;
  font-weight: bolder;
  text-align: center;
  letter-spacing: 2px;
  display: block;
  font-family: 'GO';
}
#sliderGrades #gradesList{
  font-size: 20px;
}
#slider #sliderGrades .fa-star,#slider #sliderGrades .fa-star-o{
  color: #fad82d;
  margin-right: 3px;
  font-size: 18px;
}
#slider #sliderGrades #voters{
  font-size: 13px;
  display: block;
  margin-bottom: 15px;
  text-align: right;
}
#sliderGrades .fa-clock-o{
  margin-right: 5px;
}
#sliderGrades #sliderTime,#sliderGrades #sliderGenres,#sliderGrades #voters{
  color: #dbced9;
}
#slider #sliderCircles{
  width: 5%;
  justify-content: space-between;
  margin: 0px auto;
}
#slider #sliderCircles .fa-circle, #slider #sliderCircles .fa-circle-o{
  font-size: 15px;
  color: #fad9f8;
}
.disabledReservation{
  opacity: 0.5;
}
#movies, #centeredWrapper, #resWrapper, #centeredComingSoon,
#centeredMembershipBg, #centeredAboutCinema,
#centeredAboutAuthor, #centeredContactAdmin, #centeredContactCinema,
#centeredAccountAccessBg, #adminMovies,#adminProjections,
#adminUsers,#commentsLinksAdmin, #activity{
  background-color: #1f1a19;
  color: #dbced9;
  padding: 35px 40px;
  width: 80%;
  margin: 0px auto;
}
#movies #headlineSearchCont{
  display: flex;
  justify-content: space-between;
}

```

```

}
#movies #headlineDecorationalMovies{
  width: 80%;
  /*float: left;*/
}
#movies #headlineSearchCont h2, #centeredWrapper h2, #resWrapper h2,
#centeredComingSoon h2, #centeredMembership h2,
#centeredAboutCinema h2, #centeredAboutAuthor h2,
#centeredContactAdmin h2, #centeredContactCinema h2, #centeredAccountAccess h2,
#centeredAdmin h2{
  text-align: center;
  font-size: 30px;
  letter-spacing: 2px;
  font-weight: 300;
}
.decorationalDiv{
  position: relative;
  display: flex;
}
.decorationalDiv::before, .decorationalDiv::after{
  content: '';
  display: block;
  border-bottom: 1px solid #666;
  -webkit-flex-grow: 1;
  -webkit-box-flex: 1;
  -moz-flex-grow: 1;
  -ms-flex-positive: 1;
  flex-grow: 1;
  -webkit-flex-shrink: 1;
  -webkit-box-flex: 1;
  -moz-flex-shrink: 1;
  -ms-flex-negative: 1;
  flex-shrink: 1;
  -webkit-flex-basis: auto;
  -moz-flex-basis: auto;
  -ms-flex-preferred-size: auto;
  flex-basis: auto;
  height: 1px;
}
.decorationalDiv span{
  display: flex;
  margin-top: -8px;
  padding: 0px 5px;
}
.decorationalDiv span:after{
  content: '\25C7';
  color: #a0aec0;
}
#movies #searchCont{
  position: relative;
}
#movies .searchicon {
  font-size: 23px;
  float: left;
  margin-top: -20px;
  /*margin-left: 80px;*/
  position: relative;
  top: 22px;
  left: 30px;
}

```

```

        color: #bdb1bb;
    }
    #movies .searchicon:hover{
        cursor: pointer;
    }
    #movies #extendSearch{
        border-width: 0px;
        color: #dbced9;
        font-size: 13px;
    }
    .extend {
        display: none;
        width: 0%;
        box-sizing: border-box;
        border-width: 0px;
    }
    .extended { /*:focus*/
        /*display: initial;*/
        box-sizing: border-box;
        outline: 0;
        border-bottom: 2px solid #666;
        background-image: linear-gradient(#0f0d0c, #1f1a19);
        width: 80%;
        padding: 7px 0px 7px 32px;
        -webkit-transition: width 3.4s ease-in-out;
        transition: width 0.4s ease-in-out;
    }
    #movies .fa-close{
        position: absolute;
        right: 12%;
        top: 10%;
        font-size: 13px;
    }
    #movies .fa-close:hover{
        cursor: pointer;
    }
    #movies #filterSortMovies{
        /*width: 85%;*/
        width: 100%;
        margin-top: 30px;
        display: flex;
        justify-content: space-between;
    }
    #movies #filterSortMovies .ddlFilter{
        width: 17%;
        padding: 7px 2px 7px 7px;
        background-image: linear-gradient(#0f0d0c, #1f1a19);
        font-size: 16px;
        border-width: 0px;
        border-bottom: 1px dotted #525050;
        border-radius: 7px;
    }
    #movies #dateFilter{
        color: #dbced9;
    }
    #movies #filterSortMovies .ddlFilter option{
        background-color: #0f0d0c;
    }
    .greyOption{

```

```

        color: #757272;
    }
    .ddlChangingOptionsColor{
        color: #dbced9;
    }
    .ddlDefaultOption{
        color: #757272;
    }
    .alwaysWhiteOptions{
        color: #dbced9 !important;
    }
    .alwaysWhiteOptionAdded{
        color: #dbced9 !important;
    }
    #deleteFiltersDiv{
        display: flex;
        margin-top: 15px;
        font-size: 14px;
        color: #777;
    }
    #deleteFiltersDiv:hover{
        cursor: pointer;
    }
    #deleteFilters{
        background-image: url('../images/delete1.svg');
        background-size: contain;
        width: 18px;
        height: 18px;
        margin-right: 3px;
    }
    #movies #filterSortMovies #sortContainer{
        font-size: 15px;
        margin-left: 5px;
        color: #757272;
    }
    #movies #filterSortMovies #sortContainer .fa-sort-alpha-asc{
        margin-right: 5px;
        font-size: 12px;
    }
    #movies #filterSortMovies #sortContainer #ascOrder, #movies #filterSortMovies
    #sortContainer #descOrder {
        margin: 0px 3px 0px 7px;
        color: #000;
    }
    #movies #moviesContainer #ticketsMeaning{
        margin-bottom: 30px;
        font-size: 15px;
        color: #858181;
        text-align: right;
    }
    #movies #moviesContainer #ticketsMeaning .fa-ticket{
        transform: rotate(45deg);
        font-size: 17px;
    }
    #movies #moviesContainer #ticketsMeaning #greenTicket{
        color: #65ab48;
        margin-right: 5px;
    }
    #movies #moviesContainer #ticketsMeaning #redTicket{

```

```

    color: #e3002a;
    margin: 0px 5px 0px 10px;
}
#movies img {
    max-width: 100%;
    height: auto;
}
#movies #moviesContainer{
    margin-top: 70px;
}
#movies #moviesContainer .movie{
    display: flex;
    justify-content: space-between;
}
#movies .picGeneralMovieInfo{
    display: flex;
    justify-content: space-between;
    width: 80%;
}
.voteDiv{
    position: absolute;
    text-align: center;
    z-index: 20;
    bottom: 40px;
    right: -60px;
}
.voteDiv .success{
    font-size: 13px;
}
.vote{
    margin-left: 5px;
    font-size: 13px;
    color: #858181;
    z-index: 2;
}
.faCloseRateCloud{
    height: 15px;
}
.giveVote{
    display: none;
    box-sizing: content-box;
    padding: 10px 20px 15px 20px;
    background-image: url('../images/chat.svg');
    background-size: contain;
    background-position: center center;
    background-repeat: no-repeat;
    height: 160px;
}
.giveVoteList{
    margin: 3px 0px;
}
.giveVoteList li .fa{
    font-size: 15px;
}
.confirmVote{
    padding: 5px;
    border: 0.5px solid #757272;
    background-color: transparent;
    letter-spacing: 1px;
}

```

```

    border-radius: 5px;
    color: #fff;
    display: block;
    margin: 5px auto 0px auto;
}
.confirmVote, .giveVoteList li .fa:hover{
    cursor: pointer;
}
.userChosenGrade{
    font-family: 'GO';
    font-size: 20px;
    letter-spacing: 2px;
}
#movies .moviePic{
    width: 35%;
    position: relative
}
/*#movies .moviePic .movieAbsolute{*/
/*    width: 100%;*/
/*    height: 100%;*/
/*    position: absolute;*/
/*    background-color: rgba(0, 0, 0, 0.7);*/
/*}*/
#movies .moviePic img{
    box-shadow: 0px 5px 20px #000;
}
/*#movies .movieInfo{*/
/*    width: 70%;*/
/*    display: flex;*/
/*    justify-content: flex-start;*/
/*}*/
#movies .generalMovieInfo{ /*.movieInfo .generalMovieInfo*/
    width: 60%;
}
#movies h3{ /*.movieInfo*/
    font-size: 25px;
    letter-spacing: 1px;
    font-weight: normal;
    margin-bottom: 10px;
    display: flex;
}
.agelimit{ /*.movieInfo*/
    display: block;
    width: 30px;
    height: 30px;
    font-size: 15px;
    border-radius: 50%;
    padding: 4px 2px 0px 0px;
    border: 1px solid #fff;
    background-image: linear-gradient(#2d0c59, #6b18d9);
    letter-spacing: 0px;
    margin: 5px 0px 0px 20px;
    text-align: center;
}
#movies .movieGrade{ /*.movieInfo*/
    display: flex;
    justify-content: space-between;
    width: max-content;
    margin: 0px 0px 10px 0px;
}

```

```

    position: relative;
}
.numberMovieGrade{ /*.movieInfo*/
    font-family: 'G0';
    font-size: 25px;
    letter-spacing: 2px;
    margin-right: 10px;
}
#movies .fa-star-o, #movies .fa-star, #movies .fa-star-half-o{
    color: #fad82d;
    margin-right: 1px;
}
#movies .movieDesc{
    margin: 30px 0px 50px 0px;
    padding-right: 20px;
}
#movies .movieDate, #movies .fa-clock-o, #movies .movieTechn{
    color: #757272;
}
#movies .movieDate span{
    color: #fff;
}
.detailsLink{
    display: block;
    width: max-content;
    /*background-image: linear-gradient(#2d0c59, #6b18d9);*/
    margin-top: 30px;
    padding: 5px;
    font-size: 11px;
    text-transform: uppercase;
    border-radius: 10%;
    position: relative;
}
.detailsLink:before {
    position: absolute;
    left: -2px;
    right: -2px;
    top: 0px;
    bottom: 0px;
    border: 1px solid #6b18d9;
    content: '';
}
.detailsLink:after{
    position: absolute;
    left: 3px;
    right: 3px;
    top: -5px;
    bottom: -5px;
    border: 1px solid #6b18d9;
    content: '';
}
#movies .movieProjections{
    margin-top: 70px;
    font-size: 15px;
    width: 30%;
    margin-left: 20px;
}
#movies .movieProjection{
    float: left;

```

```

    min-height: 100px;
    width: 33.33%;
}
#movies .movieTechn{
    padding-left: 10px;
}
#movies #moviesContainer .movie .fa-ticket{
    margin-left: 5px;
    font-size: 16px;
    transform: rotate(45deg);
}
#movies #moviesContainer .movie .greenTicket{
    color: #65ab48;
}
#movies #moviesContainer .movie .redTicket{
    color: #e3002a;;
}
#noResult{
    text-align: center;
    color: #757272;
    font-size: 23px;
    margin-bottom: 50px;
}
#pagination{
    width: max-content;
    margin: 0px auto;
}
#pagination li a{
    color: #757272;
    margin-right: 5px;
}
.activePagination{
    border-bottom: 1px dotted #757272;
}
#movies .decorationalMovieDiv{
    height: 4px;
    background-image: linear-gradient(to right,#0f0d0c,#1f1a19);
    margin: 20px 0px;
}
.decorationalFooterDiv{
    width: 80%;
    margin: 0px auto;
    display: flex;
    height: 20px;
}
.decorationalFooterDiv span:before{
    content: '\25C7';
    color: #a0aec0;
    display: block;
    flex-shrink: 1;
    flex-basis: auto;
    height: 1px;
}
.decorationalFooterChild{
    height: 1px;
    width: 100%;
    background-color: #666;
}
#decorationalFooterChildBottom{

```



```

    margin-top: 20px;
}
#footerContainer{
    margin: 20px 0px;
}
#footerInfoCont{
    display: flex;
    justify-content: space-around;
    padding:0px 30px;
}
#footer{
    width: 80%;
    padding: 0px 40px;
    border-left: 1px solid #666;
    border-right: 1px solid #666;
    margin: 0px auto;
}
#footer ul{
    display: initial;
}
#footer .footerInfo ul li a{
    font-size: 15px;
    color: #d9cacc;
}
#footerLogo{
    margin-right: 20px;
}
#footer .footerInfo h4{
    width: max-content;
    font-weight: normal;
    margin-bottom: 10px;
    letter-spacing: 1px;
    color: #d9cacc;
    text-align: center;
    border-bottom: 1px solid #fcacbb;
}
#footerContainer #copyright{
    font-size:14px;
    text-align: center;
    margin-top: 30px;
    color: #666;
}
#logoSocialsDiv{
    margin-top: 30px;
}
#logoSocialsDiv #followUs{
    margin-top: 30px;
    margin-bottom: 20px;
    text-transform: uppercase;
    letter-spacing: 1px;
    font-size: 13px;
    color: #757272;
}
#logoSocialsDiv .fa-heart-o{
    margin-left: 5px;
    color: #e3002a;
}
#footer #footerSocials{
    display: flex;

```

```

    width: 45%;
    justify-content: space-between;
}
#footer #footerSocials .fa{
    font-size: 23px;
    color: #d9cacc;
}
#footerInfos{
    display: flex;
    justify-content: space-between;
    width: 55%;
}

/* MOVIE    DETAILS    PAGE */

#backgroundPic{
    /*height: 500px;*/
    background-size: cover;
    background-repeat: no-repeat;
    background-image: url("../images/tenet5.jpg");
    position: relative;
    /*background-attachment: fixed;*/
    box-shadow: 0 15px 15px #0f0d0c;
    /*background-position: center center;*/
}
#backgroundPic:after{
    content: "";
    width: 100%;
    height: 217px;
    position: absolute;
    bottom: 0;
    left: 0;
    display: block;
    background: url('../images/page-cover-shadow.png') bottom repeat-x;
}
#transparentDivOnBg{
    z-index: 1;
    background-color: rgba(0, 0, 0, 0.8);
    height: 100%;
    position: relative;
    backdrop-filter: blur(2px);
    overflow: hidden;
}
.fancyTitle{
    display: flex;
    justify-content: center;
    padding: 30px 0px 25px 0px;
    width: 80%;
    margin: 0px auto;
}
.fancyTitlePre, .fancyTitlePost{
    position: relative;
    width: 35%;
    display: flex;
}
.fancyTitlePre::before{
    position: absolute;
    bottom: 40%;

```

```

    right: 50px;
    font-size: 20px;
    color: #999;
    content: "o";
    height: 40%;
    line-height: 1;
    transform: rotate(180deg);
}
.fancyTitlePre::after{
    font-size: 20px;
    color: #999;
    position: absolute;
    bottom: 5px;
    right: 50px;
    content: "o";
    height: 50%;
    line-height: 1.1;
}
.line{
    position: relative;
    display: -webkit-flex;
    display: flex;
    -webkit-flex-grow: 1;
    display: -webkit-box;
    /*width: 95%;*/
    height: 1px;
    background-color: #3b3b3b;
    margin-top: 20px;
}
.line:before{
    content: '';
    width: 100%;
    margin: auto;
    /* height: 1px; */
    border-bottom: 1px solid #3b3b3b;
}
.fancyTitlePre .end, .fancyTitlePost .end{
    width: 50px;
    display: -webkit-flex;
}
.fancyTitlePre .end:after{
    content: "◇";
    transform-origin: center center;
    line-height: 0;
    transform: scale(2.5, 1);
    color: rgb(119, 119, 119);
    margin: auto auto auto 9px;
    padding: 0px;
}
.fancyTitlePost::before {
    position: absolute;
    bottom: 40%;
    left: 50px;
    font-size: 20px;
    color: #999;
    content: 'o';
    height: 50%;
    line-height: 1;
    transform: rotate(180deg);
}

```

```

}
.fancyTitlePost::after {
  font-size: 20px;
  color: #999;
  position: absolute;
  bottom: 5px;
  left: 51px;
  content: 'o';
  height: 50%;
  line-height: 1.1;
}
.fancyTitlePost .end::before {
  content: '◇';
  transform-origin: center;
  margin: auto;
  margin-right: 9px;
  padding: 0;
  transform: scale(2.5, 1);
  color: #777;
  line-height: 0;
}
.fancyTitleInner h2{
  font-weight: 300;
  letter-spacing: 1px;
  color: #dbced9;
  font-size: 32px;
}
#abtMovie{
  padding-bottom: 40px;
}
#movieInformation{
  width: 70%;
  display: flex;
  justify-content: space-between;
  margin: 0px auto;
}
#leftMovieInformation{
  height: 0%;
  width: auto;
  font-size: 14px;
  color: #948b93;
  position: relative;
}
#leftMovieInformation .vote{
  display: block;
  margin-bottom: 3px;
}
#leftMovieInformation .voteDiv .fa-close{
  display: block;
  float: right;
}
#leftMovieInformation .voteDiv{
  margin: 0px;
  display: flex;
}
#leftMovieInformation .giveVoteList .fa{
  color: #fad82d;
}
#leftMovieInformation img{

```

```

    max-width: 100%;
    height: auto;
    box-shadow: 0px 5px 20px #000;
    margin-bottom: 5px;
}
#leftMovieInformation .detailsLink{
    color: #948b93;
    display: block;
    margin: 20px auto 0px auto;
}
#rightMovieInformation{
    width: 60%;
    height: 0%;
}
#rightMovieInformationTransparent{
    background: rgba(0, 0, 0, 0.5);
    /*backdrop-filter: blur(8px);*/
    width: 100%;
    height: 100%;
    padding: 20px;
    /*backdrop-filter: blur(2px);*/
    /*overflow: hidden;*/
}
.fancy-box__inner{
    padding: 20px;
}
.fancy-box{
    position: relative;
}
.fancy-box::before{
    position: absolute;
    top: 1px;
    left: 10px;
    right: 10px;
    border-top: 1px solid #666;
    content: '';
}
.fancy-box::after{
    position: absolute;
    bottom: 1px;
    left: 10px;
    right: 10px;
    border-top: 1px solid #666;
    content: '';
}
.fancy-box .fancy-box__tl {
    position: absolute;
    top: 1px;
    left: 1px;
}
.fancy-box .fancy-box__tl::before {
    position: absolute;
    left: 0;
    top: 0;
    width: 5px;
    height: 8px;
    border-top: 1px solid #666;
    border-right: 1px solid #666;
    border-bottom: 1px solid #666;
}

```

```

        content: '';
    }
    .fancy-box .fancy-box__tl::after {
        position: absolute;
        top: 0;
        left: 0;
        width: 8px;
        height: 5px;
        border-left: 1px solid #666;
        border-bottom: 1px solid #666;
        border-right: 1px solid #666;
        content: '';
    }
    .fancy-box .fancy-box__tr {
        position: absolute;
        top: 1px;
        right: 1px;
    }
    .fancy-box .fancy-box__tr::before{
        position: absolute;
        right: 0;
        top: 0;
        width: 5px;
        height: 8px;
        border-top: 1px solid #666;
        border-left: 1px solid #666;
        border-bottom: 1px solid #666;
        content: '';
    }
    .fancy-box .fancy-box__tr::after {
        position: absolute;
        top: 0;
        right: 0;
        width: 8px;
        height: 5px;
        border-right: 1px solid #666;
        border-bottom: 1px solid #666;
        border-left: 1px solid #666;
        content: '';
    }
    .fancy-box .fancy-box__bl {
        position: absolute;
        bottom: 1px;
        left: 1px;
    }
    .fancy-box .fancy-box__bl::before {
        position: absolute;
        left: 0;
        bottom: 0;
        width: 5px;
        height: 8px;
        border-bottom: 1px solid #666;
        border-right: 1px solid #666;
        border-top: 1px solid #666;
        content: '';
    }
    .fancy-box .fancy-box__bl::after {
        position: absolute;
        bottom: 0;

```

```

    left: 0;
    width: 8px;
    height: 5px;
    border-left: 1px solid #666;
    border-top: 1px solid #666;
    border-right: 1px solid #666;
    content: '';
}
.fancy-box .fancy-box__br {
    position: absolute;
    bottom: 1px;
    right: 1px;
}
.fancy-box .fancy-box__br::before {
    position: absolute;
    right: 0;
    bottom: 0;
    width: 5px;
    height: 8px;
    border-bottom: 1px solid #666;
    border-left: 1px solid #666;
    border-top: 1px solid #666;
    content: '';
}
.fancy-box .fancy-box__br::after {
    position: absolute;
    bottom: 0;
    right: 0;
    width: 8px;
    height: 5px;
    border-right: 1px solid #666;
    border-top: 1px solid #666;
    border-left: 1px solid #666;
    content: '';
}
.fancy-box .fancy-box__sides::before {
    position: absolute;
    top: 10px;
    bottom: 10px;
    left: 1px;
    border-left: 1px solid #666;
    content: '';
}
.fancy-box .fancy-box__sides::after {
    position: absolute;
    top: 10px;
    bottom: 10px;
    right: 1px;
    border-right: 1px solid #666;
    content: '';
}
#tableInfo{
    width: 100%;
    margin: 0px auto;
    color: #dbced9;
}
#tableInfo .vote{
    margin-top: 7px;
}

```

```

#tableInfo .voteDiv{
  z-index: 1;
  display: flex;
}
#tableInfo tr td:nth-child(1){
  letter-spacing: 1px;
  font-weight: 400;
  font-family: 'Rochester';
}
#tableInfo .giveVote{
  height: 180px;
  border-top: 1px solid #000;
}
#tableInfo td{
  vertical-align: middle;
  padding: 3px;
}
.ratingTd{
  display: flex;
}
.ratingTd ul{
  height: 0%;
  margin-top: 5px;
}
/*#tableInfo .voteDiv ul{*/
/*  height: initial;*/
/*}*/
.ratingTd .fa{
  color: #fad82d;
  margin-right: 2px;
}
.numberMovieGradeMovDet{
  font-size: 28px;
}
#tapeHolder{
  display: flex;
  justify-content: space-between;
  position: absolute;
  width: 100%;
  bottom: 0;
  /*overflow: hidden;*/
}
.tape{
  width: 50%;
}
#tapeHolder .tapeHold{
  /*width: 50%;*/
  height: 150px;
  background-position: center;
  background-repeat: no-repeat;
  background-image: url("../images/movie_tape.png");
}
#tapeHolder .tape:nth-child(1) .tapeHold{
  background-size: cover;
  background-position: right;
  transform: rotate(50deg);
  width: 50%;
  margin-left: -80px;
}

```



```

#tapeHolder .tape:nth-child(2) .tapeHold{
  float: right;
  background-position: right;
  transform: rotate(-10deg);
  width: 60%;
  background-size: cover;
  margin-right: -20px;
}
#centeredWrapper{
  background-image: linear-gradient(to top, #1a1614, #1f1a19);
}
#projectionsMovie{
  margin-top: 30px;
}
.sparkleCam{
  display: flex;
}
.cameraDecorationHeadline{
  width: 160px;
  height: 160px;
  background-size: contain;
  background-image: url("../images/movie-camera.svg");
  margin-top: 50px;
}
.sparkle{
  width: 40px;
  height: 40px;
  background-size: contain;
  background-image: url("../images/sparkleGrey.svg");
  margin-top: 75px;
}
#rotatedCam{
  transform: rotateY(180deg);
}
#movProjHolder{
  margin-top: 40px;
  width: 45%;
}
#movProjHolder .movieProjection{
  width: 30%;
  float: left;
  min-height: 90px;
  margin-right: 10px;
  margin-bottom: 20px;
  padding-left: 50px;
}
#movProjHolder .movieProjection .movieTechn{
  color: #757272;
}
#movProj{
  display: flex;
  justify-content: space-between;
  margin-top: 20px;
}
#movProjHolder .fa-ticket{
  transform: rotate(45deg);
  font-size: 17px;
}
#movProjHolder .redTicket{

```

```

        color: #e3002a;
    }
    #movProjHolder .greenTicket{
        color: #65ab48;
    }
    #plotText{
        padding: 20px 30px;
        text-align: justify;
        font-size: 14px;
        -webkit-columns: 2;
        -moz-columns: 2;
        columns: 2;
        column-gap: 40px;
        color: #dbced9;
    }
    #comments{
        width: 80%;
        margin: 0px auto;
        color: #dbced9;
        font-weight: normal;
        background-image: url('../images/form-pattern.jpg');
        background-size: contain;
        background-attachment: fixed;
    }
    #commentsTransparent{
        /*background-color: #3b3130;*/
        background-color: rgba(15, 13, 12, 0.9);
        padding: 25px;
        backdrop-filter: blur(1px);
    }
    #numOfComments{
        font-size: 13px;
    }
    #numOfComments::before, .comment .usernameComment::after, .editedComm::before,
    .replyComment .usernameComment::after{
        content: 'o';
        margin: 0px 7px;
        font-size: 12px;
        font-weight: 100;
        color: #757272;
    }
    #commentsTransparent .fa-comment{
        margin-right: 7px;
        font-size: 15px;
    }
    #peopleComments{
        padding: 25px 20px;
        width: 90%;
        margin: 0px auto;
    }
    #leaveComm{
        display: flex;
        padding-bottom: 10px;
    }
    #leaveComm #myComment, .updateCommArea, .myCommentReply, .myCommentReply2{
        width: 100%;
        background-color: #0f0d0c;
        border-width: 0px;
        border-bottom: 1px solid #948b93;
    }

```

```

    color: #dbced9;
    font-size: 15px;
    padding: 7px 5px 0px 5px;
}
.comment, .topPartComment, .bottomPartComment{
    display: flex;
}
.commentReplyHolder{
    margin-bottom: 25px;
}
#leaveComm{
    margin-bottom: 25px;
}
.comment, #leaveComm, .replyComment{
    background-image: linear-gradient(#0f0d0c, rgba(0,0,0,0.4));
    /*border-bottom: 2px dotted #0f0d0c;*/
    border-bottom: 1px solid #292321;
    border-left: 1px solid #292321;
}
#leaveComm .fa-pencil, .comment .fa-close{
    display: flex;
    align-items: center;
    padding-right: 10px;
    font-size: 15px;
}
.comment .fa-close{
    margin-top: -10px;
}
.userFaHolder{
    display: flex;
    flex-direction: column;
    align-items: center;
    padding-left: 5px;
    padding-top: 12px;
}
#comments .fa-user{
    font-size: 30px;
    padding: 0px 0px 7px 7px;
}
.memberSince, .memberDate{
    font-size: 9px;
    white-space: nowrap;
}
.memberSince{
    color: #948b93;
}
.commentRight{
    width: 100%;
    padding: 0px 10px 0px 15px;
    word-wrap: break-word;
}
#comments .maleUser{
    color: #a4b0de;
}
#comments .femaleUser{
    color: #e0a6d6;
}
.usernameComment{
    font-size: 16px;
}

```

```

}
.commPosted, .editedComm{
    font-size: 13px;
    margin-top: 3px;
    color: #948b93;
}
.commText{
    padding: 10px 0px 7px 5px;
}
.commentTextDiv{
    width: 100%;
}
.topPartComment{
    padding-top: 3px;
}
.bottomPartComment{
    font-size: 14px;
}
.bottomPartComment .fa{
    font-size: 14px;
    padding: 3px 7px;
    height: 0%;
}
.bottomPartComment .fa:hover, #leaveComm .fa-pencil:hover{
    cursor: pointer;
}
.adminComment{
    font-size: 14px;
    padding-bottom: 2px;
}
.bottomPartComment .angle{
    border-right: 1px solid #999999;
}
.repply{
    padding: 3px 7px;
}
.repply:hover{
    cursor: pointer;
}
.repliesDiv{
    width: 90%;
    display: flex;
    flex-direction: column;
    margin-left: auto;
}
.newRepply{
    display: flex;
    justify-content: space-between;
    padding: 10px;
    background-image: linear-gradient(#0f0d0c, rgba(0,0,0,0.4));
    border-bottom: 1px solid #292321;
}
#viewMoreCom{
    display: block;
    text-align: center;
    font-size: 14px;
}
.commentRightRepply{
    width: 100%;
}

```

```

padding: 5px 7px;
}
.repplyComment{
display: flex;
}
.repplyComment .fa-close{
padding: 5px;
}

/* END MOVIE DETAILS PAGE */

/* S T Y L E S E L E C T */

/* The container must be positioned relative: */
.custom-select {
position: relative;
width: 27%;
border-bottom: 1px solid #757272;
background-image: linear-gradient(to top, transparent 49%, #000 49%, #000 51%, transparent 51%);
height: 0%;
/*font-family: Arial;*/
}

.custom-select select {
display: none; /*hide original SELECT element: */
}

.select-selected {
background-color: transparent;
}

/* Style the arrow inside the select element: */
.select-selected:after {
position: absolute;
content: "";
top: 14px;
right: 10px;
width: 7px;
height: 7px;
border: 1px solid red;
-webkit-transform: rotate(45deg);
-moz-transform: rotate(45deg);
-ms-transform: rotate(45deg);
transform: rotate(45deg);
/*border: 6px solid transparent;*/
border-color: transparent #757272 #757272 transparent;
}

/* Point the arrow upwards when the select box is open (active): */
.select-selected.select-arrow-active:after {
border-color: #757272 transparent transparent #757272;
top: 17px;
}

/* style the items (options), including the selected item: */
.select-items div,.select-selected {
color: #757272;
padding: 8px 16px;
}

```

```

border: 1px solid transparent;
border-color: transparent transparent rgba(0, 0, 0, 0.1) transparent;
cursor: pointer;
}

/* Style items (options): */
.select-items {
    position: absolute;
    background-color: #0f0d0c;
    top: 100%;
    left: 0;
    right: 0;
    z-index: 99;
}

/* Hide the items when the select box is closed: */
.select-hide {
    display: none;
}

.select-items div:hover, .same-as-selected {
    background-color: rgba(0, 0, 0, 0.1);
}

#reservationHolder, #pricesHolder{
    width: 80%;
}
#forgetPassHolder{
    width: 60%;
}
#reservationForm{
    margin-top: 40px;
    padding: 40px 100px;
}
#chooseInfoRes, #chooseNumSeats{
    font-size: 17px;
    margin-bottom: 40px;
}
#resDdlHolder{
    display: flex;
    justify-content: space-between;
}
#resHeadlineOpacity{
    display: flex;
    justify-content: center;
    padding: 30px 0px;
    width: 100%;
    height: 100%;
}
#continueReservation{
    background-color: transparent;
    border-width: 0px;
    font-size: 15px;
    color: #757272;
    border-bottom: 1px solid #757272;
}
#continueReservation:hover{
    cursor:pointer;
}

```

```

#continueReservationDiv .fa-hand-o-right{
    margin-right: 7px;
}
#continueReservationDiv{
    display: flex;
    justify-content: flex-end;
    margin-top: 100px;
}
#alertRes, #alertForgetPass{
    color: #dbced9;
    padding: 10px 0px 10px 10px;
    background-color: #0f0d0c;
    /*background-image: linear-gradient(to top,#000, transparent, #0f0d0c);*/
}
#alertRes .fa, #alertForgetPass .fa{
    /*color: #e3002a;*/
    color: #757272;
    margin-right: 7px;
}
#blurBackgroundReservation , #blurBackgroundPrices, #blurBackgroundForgetPass,
.blurBackgroundsAdmin{
    display: none;
}
#chooseNumSeats{
    margin-top: 40px;
}
#numSeats{
    margin-left: 10px;
    background-image: linear-gradient(#000, #0a0a09);
    color: #757272;
    border-width: 0px;
    border-bottom: 1px solid #000;
    font-size: 15px;
    padding: 7px 10px;
}
#inputSeatsNum{
    padding: 5px 0px 5px 7px;
    background-image: linear-gradient(#0f0d0c, #1f1a19);
    color: #757272;
    border-width: 0px;
    border-bottom: 1px solid #0f0d0c;
}
#projectionInfo{
    display: block;
    margin: 20px 0px 0px 10px;
}
#projectionInfo span{
    color: #d9cacc;
    font-size: 14px;
    font-weight: bold;
    letter-spacing: 1px;
    margin-right: 7px;
}
.notLoggedIn{
    text-align: center;
    font-size: 17px;
    padding: 25px 0px;
}
#hallSeatsHolder{

```

```

    display: flex;
    justify-content: space-between;
    margin-top: 25px;
}
#hallSeats{
    width: 50%;
    padding: 20px 20px 10px 20px;
    border-radius: 10%;
    border-bottom: 10px solid #0f0d0c;
    margin-bottom: 20px;
}
.hallRow{
    display: flex;
    width: min-content;
    margin: 0px auto 7px auto;
}
.seat{
    background-size: contain;
    background-repeat: no-repeat;
    margin-right: 10px;
    width: 50px;
    height: 50px;
    /*width: 10%;*/
    /*height: 35px;*/
}
.not_available_seat{
    background-image: url('../images/reserved_sofa.svg');
}
.available_seat{
    background-image: url('../images/available_sofa.svg');
}
.available_seat:hover{
    cursor: pointer;
}
.hallwayLeft{
    margin-left: 25px;
}
.hallwayRight{
    margin-right: 25px;
}
#screenP{
    margin-top: 25px;
    font-style: italic;
    letter-spacing: 1px;
    text-align: center;
}
#seatsChosen{
    width: 40%;
    padding: 0px 20px 0px 20px;
}
#seatsChosenList{
    display: block;
    width: auto;
    margin: 15px auto 0px auto;
}
#chosenSeatsP{
    text-align: center;
}
#selectMoreSeats{

```



```

    font-size: 14px;
}
#seatsChosenList li{
    text-align: center;
    border-bottom: 1px dotted #666;
    position: relative;
    padding: 7px 0px;
}
#seatsChosenList li .fa-close{
    margin-left: 7px;
    color: #e3002a;
    position: absolute;
    right: 10px;
    padding-top: 5px;
}
#seatsChosenList li .fa-close:hover{
    cursor: pointer;
}
#confirmReservation{
    margin: 15px auto;
    padding: 8px;
    color: #fff;
    border-radius: 5px;
    border-width: 0px;
    font-size: 14px;
    background-image: linear-gradient(#2d0c59, #6b18d9);
    display: block;
}
#resSum{
    margin-top: 15px;
}
#resSum span{
    color: #ac76f5;
}
#abtMovieBook{
    z-index: 1;
}
#centeredComingSoon, #centeredContactCinema{
    padding: 0px;
    position: relative;
}
#csHolderOpacity, #contactCinemaHolderOpacity{
    z-index: 2;
    padding: 30px 45px 25px 45px;
    background-color: rgba(0, 0, 0, 0.8);
    height: 100%;
    position: relative;
    backdrop-filter: blur(2px);
}
#curtain{
    position: absolute;
    background-size: cover;
    background-position: center;
    width: 100%;
    height: 100%;
    z-index: 1;
    background-image: url("../images/curtainpng.png");
    /*bluecurtain*/
}

```

```

#csMovieHolder{
    margin: 0px auto;
    padding: 25px 0px;
    z-index: 2;
    position: relative;
    overflow: hidden;
}
.csInfoHolder{
    margin: 0px auto;
    padding: 7px 10px;
}
.inCinemasCS span{
    color: #858181;
}
.csInfoHolder .fa-clock-o{
    color: #6b18d9;
}
.cs_movie{
    float:left;
    width: 33%;
    height: 450px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    margin-bottom: 15px;
}
.cs_movie img{
    max-width: 100%;
    height: auto;
}
#centeredMembershipBg, #centeredAccountAccessBg, #adminUsers{
    padding: 0px;
    background-image: url("../images/form-pattern.jpg");
    background-size: contain;
    background-repeat: repeat;
}
#centeredMembership, #centeredAccountAccess, #adminUsersBg{
    background-image: linear-gradient(#000, rgba(0, 0, 0, 0.9));
    padding: 35px 40px;
    padding: 35px 40px 80px 35px;
}
#membershipForm, #forgetPassForm, #accountAccessForm, #membershipRequestForm{
    width: 50%;
    margin: 50px auto 0px auto;
}
#forgetPassForm{
    padding-bottom: 35px;
}
#membershipForm #memName, #membershipForm #memLastname, #membershipForm #memEmail,
#contactAdmin #emailAddress, #contactAdmin #body, #contactCinema
#firstnameContCin,
#contactCinema #lastnameContCin, #contactCinema #emailAddressContCin,
#contactCinema #question,
#forgetPassForm #forgetPassEmail, #accountAccessForm #accountAccessEmail,
#accountAccessForm #accountAccessCode,
#accountAccessForm #accountAccessPass, #membershipRequestForm #memReqName,
#membershipRequestForm #memReqLastname,
#membershipRequestForm #memReqEmail, #membershipRequestForm

```

```

#memReqCode,.inputAdminClass{
    width: 100%;
    margin-bottom: 15px;
    padding: 10px;
    border-width: 0px;
    background-color: #0f0d0c;
    color: #757272;
    font-size: 15px;
    border-radius: 5px;
    border: 0.5px solid #757272;
    font-family: 'Nunito Sans', sans-serif;
}
#helloAccountAccess{
    padding-bottom: 15px;
}
.noticeSuccErrMembership{
    margin-top: 15px;
}
#sumbitMembership, #sendContCin, #forgetPassSubmit, #accountAccessSubmit,
#sumbitReqMembership{
    width: 100px;
    height: 35px;
    background-color: #0a0a09;
    color: #fff;
    border: 0.5px solid #757272;
    border-radius: 5px;
    font-size: 15px;
    display: block;
    margin: 10px auto 0px auto;
}
#sumbitMembership, #forgetPassSubmit, #accountAccessSubmit:hover{
    cursor: pointer;
}
#membershipNotice1{
    margin-top: 25px;
}
#membershipNotice1 span{
    color: #e3002a;
}
#membershipNotice2{
    font-style: italic;
    margin-top: 30px;
    text-align: center;
}
#membershipNotice2 .fa{
    color: #e3002a;
}
#centeredAboutCinema h2{
    margin: 15px;
}
#aboutCinemaSlider{
    height: 400px;
    background-image: url("../images/about_cinema_main.jpg");
    background-position-y: 130%;
    background-attachment: fixed;
    background-size: cover;
    background-repeat: no-repeat;
}
#abtCinemaInfo{

```

```

    column-count: 2;
    column-gap: 20px;
    margin-top: 35px;
}
#abtCinemaInfo p{
    margin-bottom: 20px;
}
.abtCinemaPic{
    max-width: 100%;
    height: auto;
    margin: 10px 0px;
}
#author, #contactAdminDiv{
    display: flex;
    justify-content: space-between;
    margin-top: 25px;
}
#picAuthor{
    width: 35%;
}
#picAuthor img{
    max-width: 100%;
    height: auto;
}
#authorAbout{
    width: 60%;
}
#authorAbout p{
    text-align: justify;
}
#adm{
    padding-top: 70px;
    font-size: 18px;
    width: 45%;
    letter-spacing: 1px;
}
#contactAdmin{
    width: 45%;
    padding: 30px;
    border-left: 2px dotted #362d2b;;
}
#deadpool{
    margin-top: 15px;
}
#sendAdmin{
    display: block;
    position: relative;
    z-index: 3;
    padding: 10px 8px;
    color: #fff;
    border-width: 0px;
    font-size: 13px;
    letter-spacing: 1px;
    border-radius: 5px;
    cursor: pointer;
    background-color: transparent;
}
#sendAdminDiv{
    width: max-content;

```

```

        margin:0px auto;
        position: relative;
    }
    #sendAdminDiv:before {
        position: absolute;
        left: -2px;
        right: -2px;
        top: 0px;
        bottom: 0px;
        border: 1px solid #e3002a;
        content: '';
    }
    #sendAdminDiv:after {
        position: absolute;
        left: 3px;
        right: 3px;
        top: -5px;
        bottom: -5px;
        border: 1px solid #e3002a;
        content: '';
    }
    .cute{
        /*box-shadow: 0px 8px 8px -5px #000;*/
    }
    #purplehero, #spearehero{
        float: right;
    }
    #contactCinemaHolder{
        width: 70%;
        margin: 35px auto 0px auto;
    }
    #contCinNotice{
        margin-top: 25px;
        text-indent: 10px;
        /*white-space: nowrap;*/
    }
    #contCinNoticeDiv{
        /*width: max-content;*/
    }
    #contCinNoticeDiv #reply{
        float: right;
        margin-top: -5px;
        margin-right: -10px;
    }
    .phoneHolder{
        display: flex;
        font-size: 14px;
    }
    .phoneHolder:nth-child(2){
        margin: 10px 0px 0px 0px;
    }
    .phoneHolder img{
        margin-right: 5px;
    }
    .phoneHolder p{
        padding-top: 7px;
    }
    #contCinSendMail{
        font-family: 'SF';
    }

```

```

    font-size: 20px;
    letter-spacing: 1px;
    padding-bottom: 15px;
    text-indent: 10px;
    color: #858181;
}
#question{
    color: #757272;
}
#nameHolderContCinema{
    display: flex;
}
#contactCinema #firstnameContCin{
    margin-right: 15px;
}
#contactCinemaHolderOpacity{
    padding-bottom: 40px;
}

/* A D M I N */

.blurBackgroundsAdmin{
    position: fixed;
    bottom: 0;
    right: 0;
    left: 0;
    top: 0;
}
#closeUpdateMovieFormHolder, #closeInsertMovieFormHolder,
#closeUpdateUserFormHolder, #closeUpdateLinkFormHolder, #closeInsertLinkFormHolder
{
    height: 40px;
}
.inputAdminClass{
    margin-bottom: 0px
}
#genresListMovUpd, #actorsListMovUpd, #actorsListMovIns, #genresListMovIns{
    font-size: 12px;
    margin-top: 5px;
}
.movUpdActGenres{
    float: left;
    margin-right: 7px;
    color: #948b93;
}
.movUpdActGenres a{
    color: #757272;
}
.movUpdActGenres .fa-close{
    margin-left: 3px;
}
.movUpdActGenres .fa-close:hover{
    cursor: pointer;
}
#updateMovieHolder, #insertMovieHolder, #updateUserHolder, #updateLinkHolder,
#insertLinkHolder{
    width: 55%;
    padding: 20px 30px;
    background-color: #0f0e0e;
}

```

```

.inputAdminClass{
    background-color: #0d0b0b;
    padding: 5px 10px;
    font-size: 13px;
}
.labelInputHolder{
    margin-bottom: 15px;
}
.labelInputHolder,.labelInputHolderMultiple{
    display: flex;
    width:100%;
}
.inputAdminLabelSecond{
    text-align:center;
}
#updateMovieForm #distributerMovUpd, #updateMovieForm #directorMovUpd,
#updateMovieForm #countryMovUpd,
#updateMovieForm #dateMovUpd, #updateMovieForm #timeMovUpd,#updateMovieForm
#nameMovUpd, #updateMovieForm #ageMovUpd,
#updateMovieForm #csMovUpd, #updateMovieForm #picMovUpd, #updateMovieForm
#picMovBgUpd, #updateMovieForm #activeSliderMovUpd,
#updateMovieForm #picSliderMovUpd,#updateMovieForm #technologiesMovUpd
#insertMovieForm #distributerMovIns, #insertMovieForm #directorMovIns,
#insertMovieForm #countryMovIns,
#insertMovieForm #dateMovIns, #insertMovieForm #timeMovIns,#insertMovieForm
#nameMovIns, #insertMovieForm #ageMovIns,
#insertMovieForm #csMovIns, #insertMovieForm #picMovIns, #insertMovieForm
#picMovBgIns, #insertMovieForm #activeSliderMovIns,
#insertMovieForm #picSliderMovIns, #insertMovieForm #technologiesMovIns,
#updateUserForm #roleUserUpd, #updateUserForm #newsUserUpd,
#updateLinkForm #textLinkUpd, #updateLinkForm #pathLinkUpd, #updateLinkForm
#footerLinkUpd, #updateLinkForm #headerLinkUpd, #updateLinkForm #parentLinkUpd,
#insertLinkForm #textLinkIns, #insertLinkForm #pathLinkIns, #insertLinkForm
#footerLinkIns, #insertLinkForm #headerLinkIns, #insertLinkForm #parentLinkIns,
/*#updateProjForm #movieProjectionUpdate, #updateProjForm
#timeProjectionUpdate,#updateProjForm #hallProjectionUpdate,*/
.linkChbHolder
{
    width: 40%;
}
#updateProjForm #movieProjectionUpdate,#updateProjForm #hallProjectionUpdate{
    width: 80%;
}
#updateProjForm #technProjectionUpdate,
#updateProjForm #resProjectionUpdate, #updateProjForm #dateProjectionUpdate,
#updateProjForm #timeProjectionUpdate{
    width: 30%;
}
.inputAdminLabel{
    width: 20%;
    color: #757272;
    font-size: 14px;
    padding-right: 5px;
}
.adminFormHeadline{
    text-align: center;
    font-size: 16px;
    letter-spacing: 1px;
    margin-bottom: 15px;
}

```

```

        font-style: italic;
    }
    .selectAdminMovHolder{
        width: 100%;
    }
    #updMov{
        display: block;
        margin: 0px auto;
        background-color: #C7181F;
    }
    #insMov{
        display: block;
        margin: 0px auto;
        background-color: #5b13ba;
    }
    #updateMovieForm select, #insertMovieForm select{
        width: 45%;
    }
    #taDetailedDesc, #taShortDesc{
        font-size: 12px;
        height: 50px;
    }
    #adminHeader{
        background-color: #0f0d0c;
    }
    #rightAdminHeaderDiv{
        display: flex;
        justify-content: flex-end;
        padding-right: 10px;
        font-size: 14px;
        color: #d6cfcf;
    }
    #rightAdminHeaderDiv .fa-heart-o{
        color: #e3002a;
    }
    #rightAdminHeaderDiv #logoutlink{
        color: #e3002a;
        margin-left: 15px;
    }
    #menuAdmin li a{
        text-transform: uppercase;
        font-size: 14px;
        letter-spacing: 1px;
    }
    #adminHeaderHolder #logoMessg{
        float: none;
        margin: 0px auto 7px auto;
        width: max-content;
    }
    #adminMoviesHolder{
        margin-top: 30px;
    }
    #extendSearchAdm{
        border-width: 0px;
        color: #dbced9;
        font-size: 13px;
    }
    #searchContAdm{
        width: 20%;
    }

```



```

    position: relative;
}
#searchContAdm .fa-close{
    position: absolute;
    right: 12%;
    top: 10%;
    font-size: 13px;
}
#searchContAdm .searchicon{
    font-size: 23px;
    float: left;
    margin-top: -20px;
    position: relative;
    top: 22px;
    left: 30px;
}
.movieAdmin{
    display: flex;
    justify-content: space-between;
    border-bottom: 1px solid #2e2625;
    padding-top: 15px;
}
.movieInfoAdmin{
    margin-bottom: 20px;
    width: 20%;
}
.movieDescAdmin{
    width: 25%;
}
.headlineAdmMovie{
    letter-spacing: 1px;
    margin-bottom: 15px;
}
.headlineAdmActors{
    margin-top: 15px;
}
.moviePicAdmin{
    font-size: 13px;
    padding-right: 25px;
}
.moviePicAdmin .fa-clock-o{
    color: #757272;
}
.movieRatingAdmin .fa{
    color: #fad82d;
    font-size: 14px;
}
.admDescP, .actorsAdmMovie{
    font-size: 14px;
    color: #757272;
}
.btnsAdminMovies {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
.btnUpdateMov, .btnDeleteMov, #insertMovie, #updMov, #insMov, #projectionsInsert,
#updProj, #updUser, #insertLink, #updLink, #insLink{

```

```

padding: 5px;
color: #fff;
border-radius: 3px;
border-width: 0px;
}
#insertMovie,#insertLink{
display: block;
margin: 15px auto 0px auto;
background-color: #5b13ba;
font-family: 'Nunito Sans', sans-serif;
}
#insertMovie .fa-plus,#insertLink .fa-plus{
padding-right: 3px;
color: #fff;
}
.btnUpdateMov, .btnDeleteMov, #insertMovie,#updMov:hover {
cursor: pointer;
}
.btnUpdateMov{
margin-bottom: 20px;
background-color: #C7181F;
}
.btnDeleteMov{
background-color: #6b18d9;
font-size: 13px;
}
.paginationAdminDiv{
margin-top: 20px;
}
.paginationAdmMov{
font-size: 14px;
}
#adminMovies #pagination li{
margin-right: 7px;
}
#updateProjHolder,#updateUserHolder, #updateLinkHolder, #insertLinkHolder{
width: 40%;
padding: 20px 30px;
margin: 50px auto;
}
#updateProjForm .errors{
margin-top: 7px;
}

/* ADMIN PROJECTIONS*/

#adminProjections{
background-image: linear-gradient(#0f0d0c, #1f1a19);
}
#adminProjections .decorationalDiv{
margin-bottom: 15px;
}
.adminProjection{
padding-bottom: 15px;
border-bottom: 1px dotted #2e2625;
}
#viewProjections{
display: none;
}

```

```

.projectionX{
  height: 15px;
  padding-top: 3px;
}
.adminProjection .fa-close{
  height: 0%;
  /*margin: auto 0px;*/
  float: right;
  color: #C7181F;
}
.adminProjection .fa-close:hover{
  cursor: pointer;
}
#projectionsNumDiv, .projectionRow{
  display: flex;
}
#projectionsNumDiv {
  justify-content: flex-start;
  margin-bottom: 15px;
  padding-left: 15px;
}
.projectionRow{
  justify-content: space-between;
}
.projectionRow .projectionLabel{
  width: 20%;
}
.projectionRow .inputAdminClass{
  margin-top: 5px;
}
.projectionRow .projectionTimeLabel, .projectionRow .projectionHallLabel{
  width: 15%;
}
.projectionRow .projectionResAvailableLabel, .projectionRow
.projectionTechnologyLabel {
  width: 10%;
}
.projectionLab{
  font-size: 13px;
  color: #d0aefc;
  white-space: nowrap;
}
#numOfProjections{
  width: initial;
  margin-left: 20px;
}
#projNumP{
  font-size: 14px;
  font-style: italic;
  color: #777;
}
#projectionsInsert, #updProj, #updUser, #updLink, #insLink{
  display: block;
  margin: 20px auto 0px auto;
  background-color: #5b13ba;
}
#seeProjections{
  font-size: 14px;
  color: #777;
}

```

```

}
#seeProjections .fa{
    color: #d0aefc;
}
#projectionViews{
    overflow-y:scroll;
    overflow-x:hidden;
    height:350px;
}
#admProj{
    width: 90%;
    margin: 0px auto;
}
.projectionViewRow,#projectionLabels{
    display: flex;
    justify-content: space-between;
}
.projectionViewRow{
    margin-bottom: 10px;
}
#projectionLabels{
    margin-bottom: 15px;
    font-size: 14px;
    color: #d0aefc;
}
.projectionViewRow .prInfo{
    font-size: 13px;
}
.projectionViewRow .prInfo,#projectionLabels label{
    width: 16.6%;
}
.lastPr {
    width: 5% !important;
}
#searchProjections{
    margin: 15px 0px 25px 0px;
    border-width: 0px;
    color: #dbced9;
    font-size: 13px;
    box-sizing: border-box;
    outline: 0;
    background-image: linear-gradient(#0f0d0c, #1f1a19);
    padding: 7px;
}
.projAdmBorder{
    height: 2px;
    background-image: linear-gradient(to right,#1f1a19,#0f0d0c);
    margin: 10px 0px;
}
#viewProjections .fa-remove{
    color: #de232b;
}
#viewProjections .fa-edit{
    margin-right: 10px;
}
#viewProjections .fa:hover{
    cursor: pointer;
}

```

```

/* ADMIN USERS */

#admUsersTableDiv{
    height: 300px;
    overflow-x:auto;
    overflow-y:auto;
    margin-top: 25px;
}

#tableUsers{
    border-collapse: collapse;
    font-size: 14px;
    width: 100%;
}

#tableUsers .fa:hover{
    cursor: pointer;
}

#tableUsers th{
    color: #d0aefc;
}

#tableUsers tr{
    padding-bottom: 7px;
    border-bottom: 1px dotted #2e2625;
}

#tableUsers td{
    padding: 5px;
}

#tableUsers th{
    padding-bottom: 7px;
}

#tableUsers .fa-remove{
    color: #C7181F;
    margin-top: 2px;
}

#adminUsersBg{
    padding-bottom: 25px;
}

.movieAdmin img{
    max-width: 100%;
    height: auto;
}

/*COMMENTS LINKS*/

#commentsLinksAdmin{
    display: flex;
    justify-content: space-between;
}

#commentsAdmin{
    width: 45%;
    padding: 0px 20px;
}

#linksAdmin{
    width: 50%;
    padding: 0px 20px;
}

#commentsAdmin .femaleUser {
    color: #e0a6d6;
}

#commentsAdmin .maleUser{

```

```

        color: #a4b0de;
    }
#commentsHolderAdmin{
    margin-top: 25px;
    height: 340px;
    overflow-y: auto;
    overflow-x: hidden;
}
.adminComment{
    margin-bottom: 10px;
    margin-right: 10px;
}
.adminComment .comment{
    padding: 10px 0px;
}
.adminComment .userFaHolder{
    padding-top: 5px;
}
.adminComment .fa-remove{
    color: #C7181F;
    padding: 3px 5px 0px 0px;
}
.adminComment .fa-remove, #linksTable .fa,#insertLink:hover{
    cursor: pointer;
}
#commentsAdmin .fa-user {
    font-size: 20px;
    padding: 0px 0px 3px 3px;
}
#tableLinksDiv{
    height: 300px;
    overflow-y: auto;
    overflow-x: auto;
}
.decorationalAdminDiv{
    width: 4px;
    background-image: linear-gradient(to top,#141111,#1f1a19);
}
#tableLinksDiv{
    margin-top: 15px;
}
#linksTable{
    width: 100%;
    font-size: 14px;
    border-collapse: collapse;
    text-align: center;
}
#linksTable th{
    color: #d0aefc;
    padding: 5px;
}
#linksTable td{
    padding: 5px;
}
#linksTable tr{
    padding-bottom: 7px;
    border-bottom: 1px dotted #2e2625;
}
#linksTable .fa-remove{

```

```

        color: #C7181F;
    }
#linksTable .fa{
    margin-right: 5px;
    margin-bottom: 5px;
}
#adminLinkUpd, #adminLinkIns{
    margin: 5px 0px 0px 10px;
}

#activity{
    background-image: linear-gradient(#0f0d0c, #1f1a19);
}
#userActivity{
    overflow-y: auto;
    overflow-x: auto;
    height: 350px;
    background-image: url("../images/popcorn-pattern.jpg");
    white-space: nowrap;
    background-repeat: repeat;
    background-size: contain;
    background-attachment: fixed;
}
#activityListHolder{
    width: 15%;
    position: relative;
    margin: 20px auto 25px auto;
}
#userActivityAbs{
    width: 100%;
    /*background-color: rgba(0,0,0,0.8);*/
    /*overflow-y: scroll;*/
    /*overflow-x: scroll;*/
}
#activityTable{
    margin: 0px auto;
    width: 100%;
    background-color: rgba(0,0,0,0.92);
    padding: 15px 25px;
}
#activityTable th{
    border: 1px solid #777;
    padding: 5px 0px;
}
#activityTable td{
    padding: 3px 3px 3px 15px;
    font-size: 14px;
    border: 1px solid #777;
}
#menuBtnDivAdm{
    display: none;
}
#top {
    position: fixed;
    width: 50px;
    height: 50px;
    background-image: linear-gradient(#000, #1f1a19);
    bottom: 24px;
    right: 40px;
}

```

```

    font-size: 2.3em;
    text-align: center;
    z-index: 8;
    display: none;
}
#top:hover{
    cursor: pointer;
}
#top .fa-arrow-up {
    font-size: 25px;
    color: #777777;
}
iframe{
    height: 25px;
    width: 115px;
}
#widget{
    background-color: #0d0b0b;
}
#widget #sfn-frame-wrapper{
    height: 100px;
    width: 100%;
}
#widget #sfn-frame-wrapper .dropzone{
    width: 100%;
}

/* RESPONSIVE */
@media screen and (max-width: 1400px){
    #logoMessg {
        width: 40%;
    }
}
@media screen and (max-width: 1340px){
    #logReg{
        width: 20%;
    }
}
@media screen and (max-width: 1280px){
    #movies #filterSortMovies{
        width: 100%;
    }
}
@media screen and (max-width: 1255px){
    #logoMessg{
        width: 45%;
    }
}
@media screen and (max-width: 1220px){
    #menuList{
        width: 100%;
    }
}
@media screen and (max-width: 1180px){
    #logReg{
        width: 25%;
    }
}
@media screen and (max-width: 1200px) {
    #movies #filterSortMovies {

```



```

        display: initial;
    }
    #movies #filterSortMovies .ddlFilter{
        margin-right: 30px;
        margin-top: 30px;
    }
    #movies #filterSortMovies #sortContainer{
        margin-top: 20px;
    }
    #movies .ageLimit {
        margin-top: -2px;
    }
}
@media screen and (max-width: 1100px){
    #movies #moviesContainer .movie,#movies .movieProjections{
        font-size:0.9em;
    }
    #movies .movieDesc {
        margin: 20px 0px 30px 0px;
    }
    #movies h3 {
        font-size: 1.5em;
    }
    #movies .ageLimit{
        width: 25px;
        height: 25px;
        font-size: 14px;
        padding: 2px 2px 0px 0px;
    }
}
@media screen and (max-width: 1050px){
    .cs_movie{
        width: 50%;
    }
}
@media screen and (max-width: 1090px){
    #logoMessg{
        width: 50%;
    }
}
@media screen and (max-width: 1030px) {
    #footerInfoCont {
        display: initial;
    }

    #footerLogo {
        width: fit-content;
        margin: 0px auto;
    }

    #followSocials {
        display: flex;
        justify-content: space-around;
        margin: 30px auto;
        width: 80%;
    }

    #logoSocialsDiv #followUs {
        margin: 0px;
    }
}

```

```

    }
    #footer #footerSocials {
        width: 30%;
    }
    #footerInfos {
        width: 70%;
        margin: 30px auto 0px auto;
    }
}
@media screen and (max-width: 1000px) {
    #logoMessg {
        width: 55%;
    }
    #movies #moviesContainer .movie {
        display: initial;
    }
    #movies .picGeneralMovieInfo{
        width: 90%;
    }
    #movies .picGeneralMovieInfo{
        margin: 0px auto;
    }
    #movies .movieProjections {
        width: 70%;
        margin: 50px auto 0px auto;
    }
    #movies .movieProjection{
        width: 25%;
    }
    #abtCinemaInfo{
        column-count: 1;
    }
    #commentsLinksAdmin{
        display: inherit;
    }
    #commentsAdmin{
        margin-bottom: 25px;
    }
    #commentsAdmin, #linksAdmin{
        width: initial;
    }
}
}
@media screen and (max-width: 945px){
    #logReg{
        width: 30%;
    }
    #logoMessg{
        width: 60%;
    }
    #hireList {
        display: none;
    }
}
}
@media screen and (max-width: 900px){
    .movieAdmin {
        display: initial;
    }
    .movieInfoAdmin{
        width: 70%;
    }
}

```

```

        margin: 0px auto 15px auto;
    }
    .movieInfoAdmin img{
        max-width: 50%;
    }
    .moviePicAdmin{
        justify-content: space-between;
        display: flex;
    }
    .underImgMovieAdm{
        margin-left: 17px;
        font-size: 15px;
        padding: 20px 0px;
    }
    .movieDateAdmin{
        margin-bottom: 7px;
    }
    .btnsAdminMovies{
        padding-bottom: 15px;
    }
    .moviePicAdmin {
        padding-right: 0px;
    }
}
@media screen and (max-width: 830px){
    #logoMessg {
        width: 65%;
    }
    #logReg{
        width: 33%;
    }
    #registrationHolder {
        width: 80%;
    }
    #searchContAdm {
        width: 40%;
    }
}
@media screen and (max-width: 825px){
    #menuList li a{
        padding-right: 20px;
    }
    #menuList > li > a:after{
        padding-left: 20px;
    }
}
@media screen and (max-width: 800px){
    .projectionRow {
        display: initial;
        margin-top: 15px;
    }
    #projectionsToInsertDiv{
        margin-top: 15px;
    }
    .projectionLabel {
        float: left;
        width: 30% !important;
        height: 65px;
        margin-right: 15px;
    }
}

```

```

    }
}
@media screen and (max-width: 781px){
  .abtCinemaInfoContainer {
    width: 100%;
  }
  #headerWrapper{
    padding-bottom: 10px;
  }
  #menuBtnDiv,#menuBtn{
    display: block;
    width: 100%;
  }
  #menu{
    width: 80%;
    background-color: #1f1a19;
    margin: 0px auto;
    position: relative;
  }
  #menuBtn{
    /*width: 80%;*/
    background-color: #0f0d0c;
    padding: 10px 0px;
    font-size: 18px;
    color: #dbced9;
    border-width: 0px;
  }
  #menuBtn .fa-list{
    color: #dbced9;
  }
  #menu{
    padding: 0px;
  }
  #menu ul li {
    display: none;
  }
  #menuList{
    display: initial;
    width: 100%;
    position: absolute;
    z-index: 5000;
  }
  .menuChildrenList {
    position: relative;
  }
  #menuBtnDiv .fa-list {
    font-size: 23px;
    padding-right: 7px;
  }

  #menu:hover #menu ul{
    position: absolute;
  }
  #menu:hover ul li {
    display: block;
    width: 100%;
    background-color: rgba(0, 0, 0, 0.95);
    border-bottom: 1px solid #666;
    text-align: center;
  }
}

```

```

        padding: 7px 0px 7px 0px;
    }
    .no-border{
        border-bottom: 0px solid red !important;
    }
    #menu ul li a{
        padding-right: 0px;
    }
    #menu ul li a:after{
        padding-left: 0px;
        content: "";
    }
}
@media screen and (max-width: 774px){
    #logReg{
        width: 35%;
    }
    .cs_movie{
        width: 100%;
    }
}

@media screen and (max-width: 768px){
    #logReg{
        width: 45%;
        margin-bottom: 10px;
    }
    #messageContainer{
        margin-left: 10px;
    }
    #logoMessg {
        float: none;
        width: 100%;
        justify-content: center;
        margin: 0px auto;
    }
    #centerSliderInfo h2{
        font-size: 40px;
    }
    #centerSliderInfo #director {
        font-size: 20px;
    }
    #sliderGrades #sliderGrade {
        font-size: 40px;
    }
    #sliderGrades #sliderTime, #sliderGrades #sliderGenres, #sliderGrades #voters
{
        font-size: 15px;
    }
    #slider #sliderActors {
        margin-top: -150px;
    }
    #slider #sliderCircles {
        width: 7%;
    }
    #movies #filterSortMovies .ddlFilter{
        width: 25%;
    }
    #followSocials{

```

```

        display: block;
        text-align: center;
    }
    #footer #footerSocials{
        margin: 10px auto 0px auto;
    }
    #footerInfos{
        display: block;
        margin: 0px auto;
    }
    .footerInfo{
        float: left;
        width: 50%;
        height: 120px;
        margin-bottom: 20px;
        text-align: center;
    }
    #footer .footerInfo h4{
        margin: 0px auto 10px auto;
    }
}
@media screen and (max-width: 730px) {
    #slider #sliderInfoContainer #sliderInfoWrapper {
        padding: 15% 0px;
    }
}
@media screen and (max-width: 670px){
    .projectionLabel {
        width: 40% !important;
    }
}
@media screen and (max-width: 650px){
    #logoHeader {
        margin-top: 15px;
        margin-bottom: 0px !important;
    }
    #menuBtnDivAdm, #menuBtnAdm {
        display: block;
        width: 100%;
    }
    #menuBtnAdm{
        background-color: #0f0d0c;
        padding: 10px 0px;
        font-size: 18px;
        color: #dbced9;
        border-width: 0px;
    }
    #menuBtnAdm .fa{
        margin-right: 5px;
    }
    #menuAdmin{
        display: initial;
        width: 100%;
        position: absolute;
        z-index: 5000;
    }
    #menuAdmin li{
        display: none;
    }
}

```

```

}
#menuAdm:hover ul li {
  display: block;
  width: 100%;
  background-color: rgba(0, 0, 0, 0.95);
  border-bottom: 1px solid #666;
  text-align: center;
  padding: 7px 0px 7px 0px;
}
#activityListHolder {
  width: 50%;
}
}
@media screen and (max-width: 630px){
  #admProj {
    width: 100%
  }
  #projectionLabels label {
    font-size: 12px;
    text-align: center;
  }
  .projectionViewRow .prInfo {
    font-size: 11px;
    text-align: center;
  }
}
@media screen and (max-width: 599px) {
  #membformDiv {
    width: 80%;
  }
  #slider {
    display: none;
  }
  #logFormDiv {
    box-shadow: 0px 1px 4px black;
  }
  #headerWrapper {
    box-shadow: 0px 10px 10px #0f0d0c;
    position: relative;
  }

  #movies #filterSortMovies .ddlFilter {
    margin-right: 20px;
  }
  #movies .movieProjections {
    width: 100%;
  }
  #author{
    flex-direction: column;
  }
  #picAuthor{
    width: 50%;
    margin: 0px auto;
  }
  #authorAbout{
    margin-top: 15px;
    width: 100%;
  }
}

```

```

    #contactCinemaHolder{
        width: 100%;
    }
}
@media screen and (max-width: 580px) {
    #movies .moviePic{
        margin-top: 40px;
    }
}
@media screen and (max-width: 560px){
    #footer #footerSocials {
        width: 60%;
    }
}
@media screen and (max-width: 550px) {
    #logReg{
        width: 60%;
    }
}
@media screen and (max-width: 500px) {
    #registrationForm{
        padding: 40px 30px;
    }
    #searchContAdm {
        width: 75%;
        margin: 15px auto 0px auto;
    }
    .movieInfoAdmin {
        width: 100%;
    }
}
@media screen and (max-width: 460px){
    #movies #moviesContainer #ticketsMeaning span:nth-child(2){
        display: block;
        width: 100%;
    }
    #movies #moviesContainer #ticketsMeaning {
        text-align: left;
    }
    #movies #moviesContainer #ticketsMeaning #redTicket {
        margin: 0px;
    }
    .logo h3{
        font-size: 1.5em;
    }
    #projectionLabels label {
        font-size: 10px;
    }
    .projectionViewRow .prInfo {
        font-size: 8px;
    }
}
@media screen and (max-width: 450px) {
    #registrationForm{
        padding: 40px 15px;
    }
    #registrationForm #regName, #registrationForm #regPass, #registrationForm
    #regPass1, #registrationForm #email, #registrationForm #tel, #registrationForm
    #town, #registrationForm #address, #registrationForm #zip {

```



```

        font-size: 14px;
    }
    #registrationForm {
        font-size: 14px;
    }
    #messageContainer{
        display: none;
    }
    #logoHeader{
        margin-bottom: 20px;
    }
    #movies #filterSortMovies .ddlFilter {
        margin-right: 10px;
    }
    #movies .picGeneralMovieInfo{
        display: initial;
    }
    #movies .moviePic{
        width: 70%;
        margin: 0px auto;
    }
    #movies .generalMovieInfo {
        width: 100%;
        text-align: justify;
    }
    #movies h3{
        padding-left: 30%;
        margin-top: 20px;
    }
    #movies .movieGrade {
        margin: 0px auto;
    }
    #movies .movie .detailsLink {
        margin: 30px auto 0px auto;
    }
}
@media screen and (max-width: 410px) {
    .footerInfo{
        width: 100%;
        height: auto;
    }
    #movies #filterSortMovies .ddlFilter {
        display: block;
        width: 80%;
        margin-left: auto;
        margin-right: auto;
    }
    #aboutCinemaSlider{
        display: none;
    }
}
@media screen and (max-width: 400px) {
    #logReg{
        width: 80%;
    }
    #movies #headlineSearchCont {
        flex-direction: column-reverse;
    }
    #movies #searchCont {

```

```

        margin-bottom: 20px;
        height: 32px;
    }
    #movies #extendSearch{
        box-sizing: border-box;
        outline: 0;
        border-bottom: 2px solid #666;
        background-image: linear-gradient(#0f0d0c, #1f1a19);
        width: 80%;
        padding: 7px 0px 7px 35px;
    }
    #movies #headlineDecorationalMovies {
        width: 100%;
    }
    #movies .searchicon{
        margin-left: 180px;
    }
    #movies {
        padding: 25px 40px;
    }
}
@media screen and (max-width: 380px){
    #movies .movieProjection {
        width: 33.33%;
    }
    #adminProjections{
        padding: 35px 20px;
    }
    #projectionLabels label {
        font-size: 8px;
    }
}

/* RESPONSIVE MOVIE_DETAILS*/
@media screen and (max-width: 1150px){
    #movProjHolder .movieProjection {
        min-height: 110px;
    }
}
@media screen and (max-width: 985px){
    /*#movProjHolder .movieProjection {*/
    /*    width: 40%;*/
    /*}*/
    .cameraDecorationHeadline {
        width: 120px;
        height: 120px;
    }
    .sparkle {
        width: 25px;
        height: 25px;
    }
    #movProjHolder {
        width: 60%;
    }
}
@media screen and (max-width: 890px){
    .ratingTd {
        display: initial;
    }
}

```

```

    }
    .numberMovieGrade{
        /*text-align: center;*/
    }
}
@media screen and (max-width: 840px){
    #movProjHolder .movieProjection {
        width: 40%;
    }
}
@media screen and (max-width: 820px){
    #movieInformation {
        width: 90%;
    }
}
@media screen and (max-width: 810px){
    .fancyTitle{
        width: 100%;
    }
    .fancyTitleInner{
        font-size: 17px;
    }
    #contactAdminDiv{
        display:initial;
    }
    #adm{
        width: 100%;
    }
    #contactAdmin{
        width: 100%;
        border-width: 0px;
    }
    #purplehero{
        margin-bottom: 15px;
    }
    #centeredContactAdmin{
        padding-bottom: 25px;
    }
}
@media screen and (max-width: 780px){
    .sparkleCam{
        display: none;
    }
    #movProjHolder {
        width: 80%;
        margin: 40px auto 0px auto;
    }
    #movProjHolder .movieProjection {
        width: 30%;
        padding-left: 40px;
    }
}
@media screen and (max-width: 770px){
    .ratingTd ul{
        margin: 0px auto;
    }
    .ratingTd .fa {
        margin-right: 0px;
    }
}

```

```

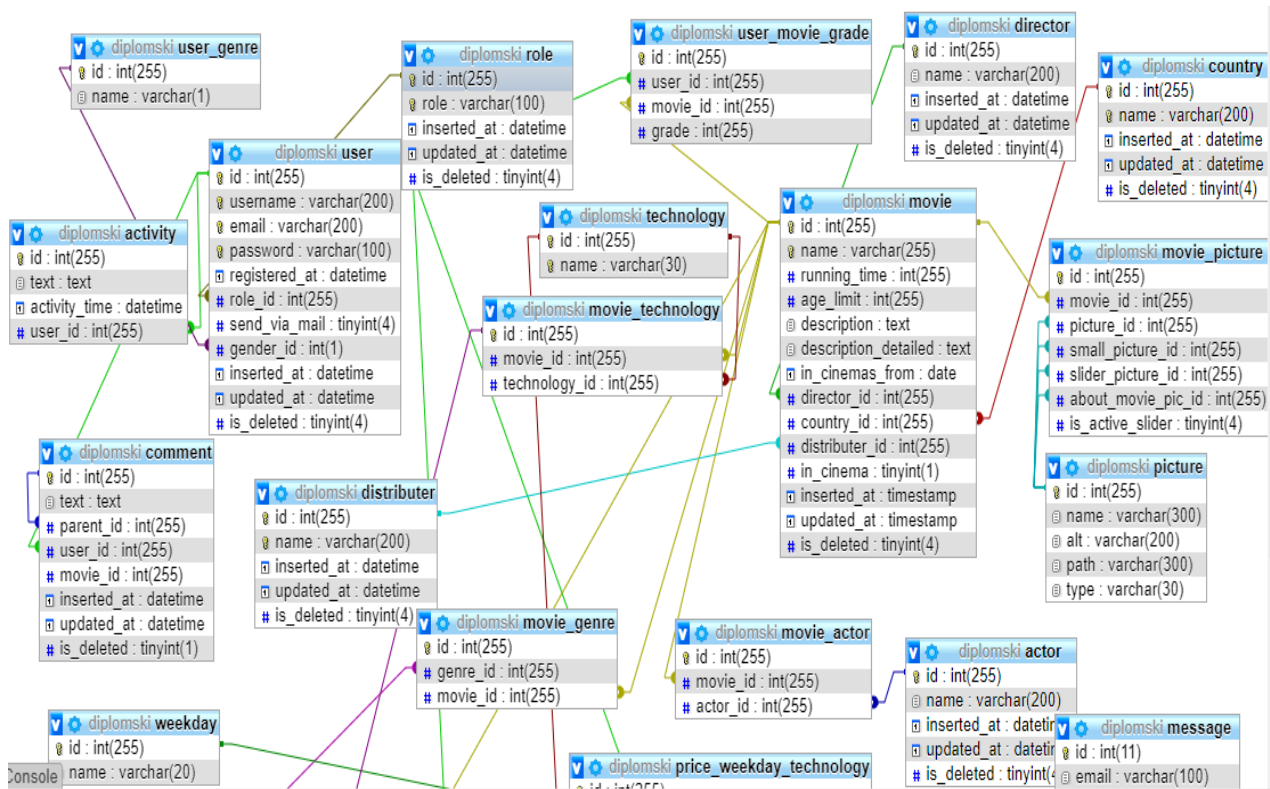
    #plotText {
        padding: 20px 10px;
        column-gap: 30px;
    }
}
@media screen and (max-width: 650px){
    .fancyTitlePre .end, .fancyTitlePost .end{
        display: none;
    }

    .fancyTitlePre::before, .fancyTitlePre::after, .fancyTitlePost::before, .fancyTitlePost::after{
        display: none;
    }
    .fancyTitleInner h2 {
        font-size: 25px;
        padding: 0px 10px;
        text-align: center;
    }
    #plotText {
        columns: 1;
    }
}
@media screen and (max-width: 620px){
    #movProjHolder {
        width: 100%;
    }
}
@media screen and (max-width: 600px){
    #movieInformation {
        display: initial;
    }
    #leftMovieInformation {
        width: 40%;
        margin: 0px auto;
    }
    #leftMovieInformation .detailsLink {
        margin-bottom: 25px;
    }
    #rightMovieInformation {
        width: 80%;
        margin: 0px auto;
    }
}
@media screen and (max-width: 580px){
    .commPosted{
        display: none;
    }
    .usernameComment::after{
        display: none;
    }
    .tapeHold{
        display: none;
    }
}
@media screen and (max-width: 480px){
    #movProjHolder .movieProjection {
        width: 40%;
    }
}

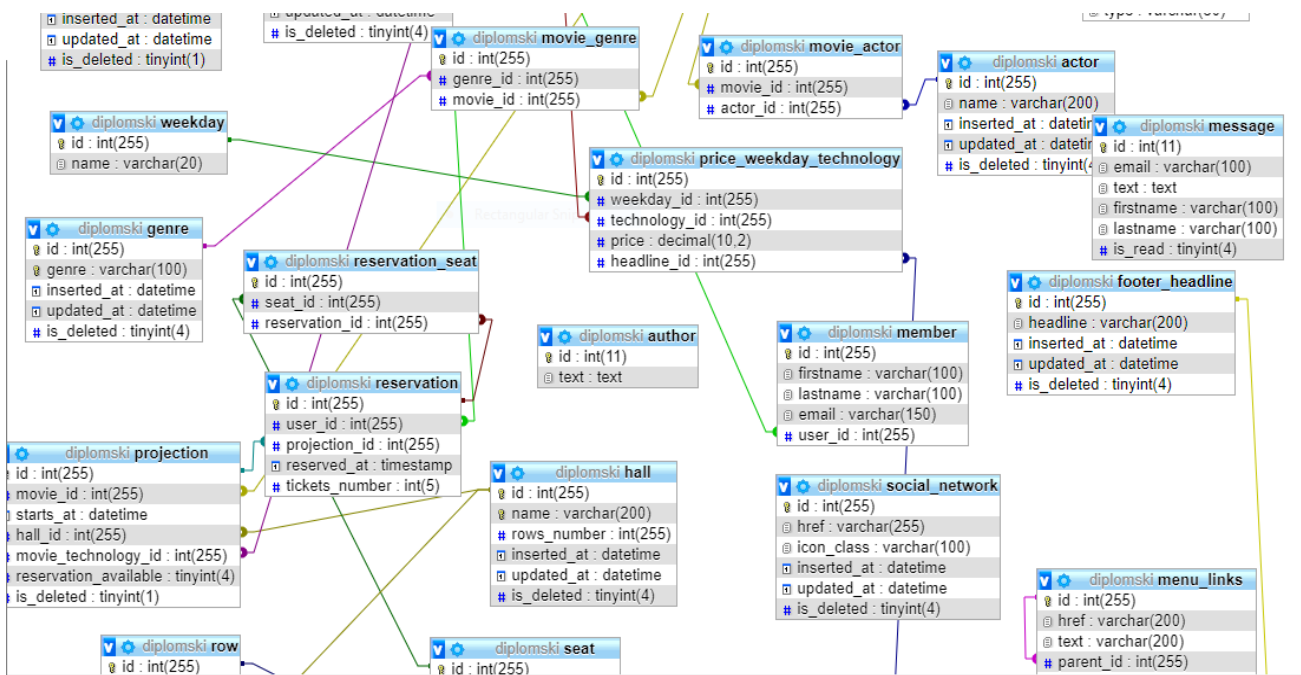
```

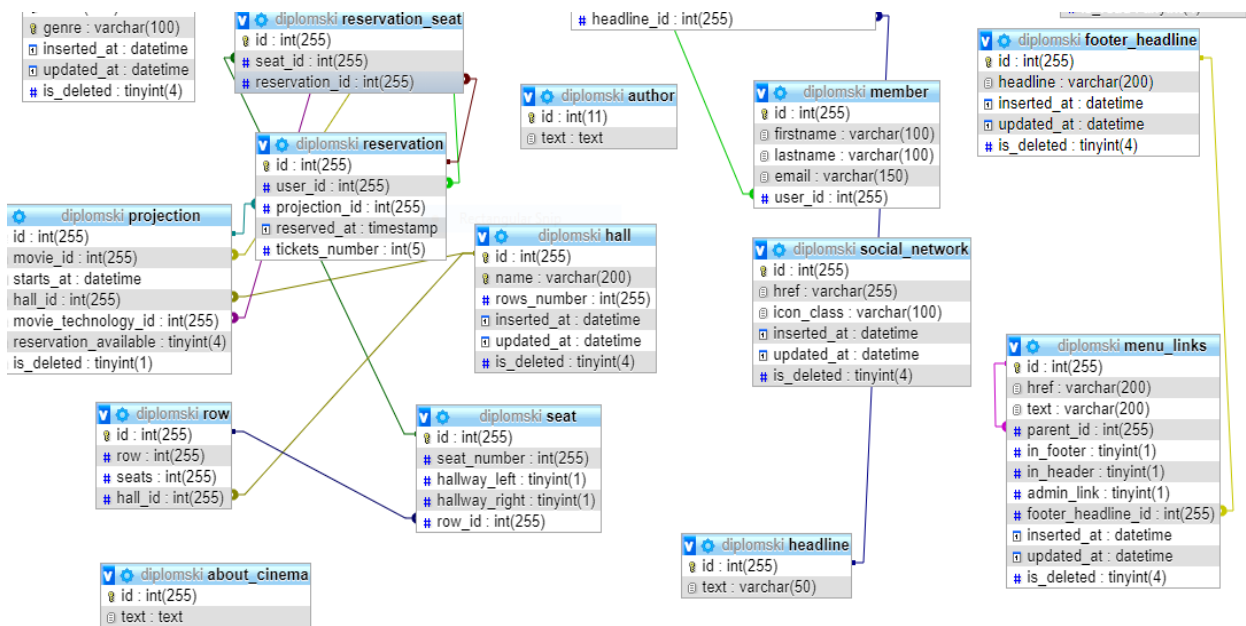
```
}  
@media screen and (max-width: 450px){  
  .userFaHolder{  
    display: none;  
  }  
  #commentsTransparent {  
    padding: 25px 15px;  
  }  
}  
@media screen and (max-width: 380px){  
  #rightMovieInformation {  
    width: 95%;  
  }  
}
```

5. Baza



Slika 4.7.1- Baza podataka





6. Zaključak

“MovieBlackout” je Web sajt za bioskop, namenjen ljubiteljima filmova i rađen je u Laravel-u. Zahvaljujući svojoj preglednosti korisnicima je omogućio lakše snalaženje i dolazak do željenih informacija.

Neautorizovani korisnik je imao mogućnost pregleda filmova na repetoaru i njihovih projekcija (neaktuelne projekcije su diskretno zatamljene), filtriranja po različitim kriterijumima i pretrage filmova.

On je mogao i da se prijavi za člansku karticu bioskopa, ostavljanjem validne email adrese, kontaktira admina ukoliko ima nekih tehničkih problema, kontaktira bioskop ukoliko ima bilo kakvih pitanja.

Autorizovani korisnik je mogao i da iskaže svoj mišljenje o filmu ostavljanjem komentara i davanjem ocene, koja kasnije ulazi u zbirnu ocenu filma.

Rezervacija karata je takođe bila moguća, klikom na aktuelnu projekciju ili klikom na stavku iz menija *Reservation* i izborom željenih opcija korisnik se redirektuje na stranicu za rezervaciju, gde bira sedišta, koja nakon uspešne rezervacije neće biti više dostupna.

Admin je uspešno manipulirao sadržajem sajta, unosio je nove, po potrebi menjao i brisao postojeće filmove i projekcije, uklanjao neželjene komentare korisnika, menjao informacije korisnika i uklanjao ih ukoliko mu se učine sumnjivim. Imao je i uredan prikaz aktivnosti admina, što mu je pružalo uvid u radnje drugih admin korisnika.

“MovieBlackout” pruža veliki broj funkcionalnosti i poseduje dobru osnovu za dalje proširivanje i razvoj. Predstavlja sigurno mesto za ljubitelje filmova, koji ovde mogu razmeniti utiske i uvek biti u toku sa najnovijim filmskim remek-delima.

7. Literatura

Laravel Application Development Cookbook Paperback - Terry Matula

Laravel: Up & Running – Matt Stauffer

Programming PHP, 4th Edition - Kevin Tatroe, Peter MacIntyre

JavaScript and JQuery: Interactive Front-End Web Development – Jon Duckett

HTML & CSS: Design and Build Web Sites – Jon Duckett

MySQL Explained: Your Step By Step Guide To Database Design - Mr. Andrew Comeau (Author), Mr. Stephen Burge (Contributor)

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://developer.mozilla.org/>

<https://laravel.com/>

<https://www.php.net/>

<https://cloudinary.com/>

<https://www.sitepoint.com/>

<https://api.jquery.com/>

<https://moderncss.dev/>