

# Web aplikacija za vođenje online videoteke

---

**Lucić, Dario**

**Undergraduate thesis / Završni rad**

**2019**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:276120>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-17**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Stručni studij**

**WEB APLIKACIJA ZA VOĐENJE ONLINE  
VIDEOTEKE**

**Završni rad**

**Dario Lucić**

**Osijek, 2019.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju**

Osijek, 22.09.2019.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju**

<b>Ime i prezime studenta:</b>	Dario Lucić
<b>Studij, smjer:</b>	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
<b>Mat. br. studenta, godina upisa:</b>	AI4565, 21.09.2018.
<b>OIB studenta:</b>	48785772977
<b>Mentor:</b>	Robert Šojo
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Doc.dr.sc. Mirko Köhler
<b>Član Povjerenstva:</b>	Doc.dr.sc. Josip Balen
<b>Naslov završnog rada:</b>	Web aplikacija za vođenje online videoteke
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rada</b>	Kreiranje web aplikacije za vođenje online videoteke u kojoj treba postojati glavni administrator i korisnici online videoteke. Administrator treba imati kompletan uvid u podatke o korisnicima videoteke, a korisnici trebaju imati uvid u filmove s cjenikom, status članarine, uvid u povijest posuđenih filmova. Omogućiti unos filmova u bazu, brisanje filmova, posudbe filmova, vođenje statistike posudbe filmova, prijava korisnika, status korisnika, statistika korisnika. Korisnici se trebaju samostalno prijaviti u videoteku, samostalnog brisanja, posuđivanje filmova, davanja ocjene filmovima. Opisati tehnologije u procesu izrade aplikacije, opisati pojedine dijelove aplikacije, te kreirati potpuno funkcionalnu aplikaciju.
<b>Prijedlog ocjene pismenog dijela ispita (završnog rada):</b>	Vrlo dobar (4)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene mentora:</b>	22.09.2019.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

## IZJAVA O ORIGINALNOSTI RADA

Osijek, 27.09.2019.

**Ime i prezime studenta:**

Dario Lucić

**Studij:**

Preddiplomski stručni studij Elektrotehnika, smjer Informatika

**Mat. br. studenta, godina upisa:**

AI4565, 21.09.2018.

**Ephorus podudaranje [%]:**

11

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za vođenje online videoteke**

izrađen pod vodstvom mentora Robert Šojo

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak završnog rada.....	1
2. TEHNOLOGIJE KORIŠTENE PRI IZRADI .....	2
2.1. HTML.....	2
2.2. CSS.....	4
2.3. Bootstrap.....	5
2.4. Javascript .....	7
2.5. PHP.....	8
3. OPIS FUNKCIONALNOSTI WEB APLIKACIJE.....	10
4. KORIŠTENJE WEB APLIKACIJE .....	18
4.1. Upute za korištenje aplikacije administratoru.....	18
4.1.1. Prijava administratora.....	18
4.1.2. Stranica administratora .....	19
4.1.3. Izbornik-korisnici .....	19
4.1.4. Dodavanje novog korisnika .....	20
4.1.5. Izbornik-filmovi .....	21
4.2. Upute korisnicima za korištenje aplikacije.....	22
4.2.1. Početna stranica i prijava .....	23
4.2.2. Videoteka .....	24
4.2.3. Gledanje filma .....	25
4.2.4. Lista gledanja .....	26
4.2.5. Povijest dodavanja u listu .....	27
4.2.6. Ocjenjivanje filma .....	28
4.2.7. Kredit i plaćanje filma .....	28

5. ZAKLJUČAK.....	30
LITERATURA.....	31
SAŽETAK.....	32
ABSTRACT.....	33
ŽIVOTOPIS.....	34
PRILOZI.....	35

# 1. UVOD

Glavni zadatak ovog rada bio je kreiranje web aplikacije za vođenje online videoteke. Aplikacija koju bi mogle koristiti dvije strane, administrator te korisnici.

Rad je podijeljen na pet poglavlja. U prvom poglavlju opisan je zadatak rada, u drugom su navedene tehnologije koje su korištene pri izradi web aplikacije, a neke od njih su HTML, CSS, Bootstrap, Javascript. Treće poglavlje opisuje funkcionalnosti aplikacije, tehničku stranu i određene dijelove kôda. Četvrto poglavlje opisuje kako aplikacija izgleda iz perspektive obje strane, kako administratora tako i korisnika te daje upute kako se koristiti njome. Detaljno su pojašnjeni koraci kako obje strane dolaze do određenih stranica.

Web aplikacija za vođenje online videoteke je prvobitno napravljena kako bi korisnici mogli gledati filmove na njoj. Administratori mogu postavljati nove filmove u ponudu, imaju uvid u korisničke aktivnosti te mogu brisati filmove iz baze. Korisnici mogu vidjeti svoju povijest gledanja, kao i dodavati nove filmove u svoju listu gledanja (engl. *Watchlist*) i ocjenjivati ih.

## 1.1. Zadatak završnog rada

Zadatak ovog završnog rada bio je uspostaviti potpuno funkcionalnu web aplikaciju za vođenje online videoteke. Nakon prijave, aplikacija preusmjerava korisnika ovisno o njegovoj ulozi na stranici. Administratori imaju veću ulogu, te mogu u potpunosti upravljati bazom podataka, te unositi nove korisnike kao i nove filmove. Njihova uloga je da obogaćuju videoteku s novim sadržajem.

## 2. TEHNOLOGIJE KORIŠTENE PRI IZRADI

U ovom poglavlju opisane su tehnologije koje su se koristile prilikom izrade ove aplikacije. Neke od njih su HTML, CSS, Bootstrap, Javascript te PHP. Svaka tehnologija korištena pri izradi bila je od iznimne važnosti te je dodala novi sloj i funkcionalnosti u aplikaciju.

### 2.1. HTML

HTML (engl. *HyperText Markup Language*) služi za izradu web stranica. Pomoću HTML jezika stvara se hipertekst dokument. HTML je opisni jezik za kreiranje HTML dokumenata ili web stranica. Bernes-Lee je čovjek koji je 1991. godine napravio HTML, a prva standardna verzija izdana je četiri godine kasnije. Godine 1999. izdana je bitna verzija, odnosno 4.01. verzija ovog opisnog jezika koja je široko korišten u svijetu, dok se trenutno koristi HTML5 koji je naslijedio prethodnu verziju 2012. godine. [2]

Kada krećemo s pisanjem kôda, on se sastoji od dva dijela:

- Glava stranice (engl. *head*)
- Tijelo stranice (engl. *body*).

Glava se sastoji od različitih metapodataka i veza koje se mogu dodati, od naslova stranice (<title>), različitih skripti (<script>) te različitih stilova koji se mogu uključiti između ključne riječi <style>. Tijelo definira dio stranice gdje se piše većina kôda. Ono se sastoji od cijelog sadržaja kojeg pišemo u HTML dokument. Od različitih tekstova, preko hipertekstualne veze, tablica, uređenih i neuređenih lista, umetanja slika i slično.

Neke od njegovih koristi:

- Navigacija internetom – pomoću određenih oznaka koje su ugrađene u ovaj jezik HTML služi kao poveznica između stranica na internetu, odnosno služi kao navigacija.
- Responzivni UI – zahvaljujući određenim razvojnim strategijama stranice HTML-a danas se mogu prikazivati na svim uređajima (mobilni uređaji, tableti, stolnim računalima i laptopima).
- Razvoj web stranica – osnovna zadaća HTML-a je razvoj strukture web stranica.
- Offline podrška – jednom učitana stranica može biti dostupna bez pristupa interneta.



- Razvoj igara - kako je HTML napredovao služi i za razvoj igara. [1]

Slika 2.1. prikazuje primjer osnovnog HTML kôd-a.

```
<!DOCTYPE html>
<html>
<head>
<title>Naziv stranice</title>
</head>
<body>

<h1>Ovo je naslov</h1>
<p>Ovo je odlomak.</p>

</body>
</html>
```

**Sl. 2.1.** *Primjer kôda jednostavne HTML stranice.*

Na slici 2.2. prikazuje primjer kôda web stranice. Kôd koji je korišten prilikom izrade login.php stranice.

```
<?php include('functions.php') ?>
<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="header">
    <h2>Login</h2>
  </div>
  <form method="post" action="login.php">
    <?php echo display_error(); ?>
    <div class="input-group">
      <label>Username</label>
      <input type="text" name="username" >
    </div>
    <div class="input-group">
      <label>Lozinka</label>
      <input type="password" name="password">
    </div>
    <div class="input-group">
      <button type="submit" class="btn" name="login_btn">Login</button>
    </div>
    <p>
      Još niste korisnik? <a href="register.php">Sign up</a>
    </p>
  </form>
</body>
</html>
```

Sl. 2.2. Primjer kôda prilikom izrade login forme.

Na slici se vidi prethodno objašnjena osnovna struktura HTML kôda, glava i tijelo. Korištena je ključna riječ <form> za kreiranje forme gdje se korisnici i administratori mogu prijaviti. Korišteni su odlomci, kao i različiti odjeljci za lakše uređivanje u CSS-u. Kao što je to u većini web stranica također je korišten i PHP, a više o njima o narednim poglavljima.

## 2.2. CSS

CSS (engl. *Cascading Style Sheet*) je opće poznat kao jednostavni stilski jezik koji služi za uređivanje HTML dokumenta, odnosno za uređivanje izgleda. On pokazuje kako bi se takav dokument trebao ispisati na ekranu. Najnovija verzija CSS-a je CSS3. [3]

Stil u HTML dokumentu može se dodavati izravno u njemu, te na način da ga se posebno uključimo glavi dokumenta. Neke od njegovih koristi:

- Brže učitavanje web stranica – Jedan CSS stil ne mora se pisati svaki put za svaku oznaku (engl. *Tag*), nego se odmah može primijeniti na sve elemente s takvom ključnom oznakom.
- CSS skraćuje vrijeme – CSS dokument se može napisati jednom i iskoristiti u više HTML dokumenata. Stil za jedan element u HTML-u može se definirati jednom i uključiti ga u koliko god HTML dokumenata je potrebno.
- Lako održavanje – Za napraviti globalnu promjenu (npr. u više datoteka) dovoljno je promijeniti jednu liniju kôda u CSS datoteci i svi elementi na web stranicama gdje je uključena ta datoteka će se automatski ažurirati. [4]

Na slici 2.3. prikazan je jednostavan primjer CSS kôda koji se dodaje izravno u HTML te se sa njime mijenja stil odlomka – pozadinsku boju u žutu.

```
<html>
  <head>
  </head>

  <body>
    <p style = "background-color:yellow;">
      This text has a yellow background color.
    </p>
  </body>
</html>
```

**Sl. 2.3.** *Primjer jednostavnog CSS kôda.*

## 2.3. Bootstrap

Bootstrap služi da bi se web stranice prilagodile npr. mobilnim uređajima te tako im omogućio responzivnost, te je namjenjen front-end programerima. On se većinom sastoji od stilski skripti. Prvobitno je razvijen od strane Twittera, a prednost je to što je potpuno besplatan i svestran. Neke od opcija Bootstrapa su dodavanje navigacija na vrhu stranice (putem navbar klase), dodavanje

velikih zaglavlja, upotreba sistema rešetke (engl. *grid system*), koji dijeli ekran na sistem s 12 kolona s mnogo već definiranih klasa itd.[5]

Bootstrap nudi velik broj CSS klasa koje se mogu jednostavno primjeniti na HTML elemente. Uz Bootstrap dolazi i velik broj jQuery dodataka (engl. *plug-ins*) koji mogu omogućiti dodatne funkcionalnosti. Kôd koji je potrebno uključiti u glavu HTML datoteke kako bi se primijenio Bootstrap na stranici prikazan je na slici 2.4. Ukoliko korisnik koristi ovakav tip primjenjivanja, bit će mu potrebna i stalna povezanost s internetom. Ako korisnik ne želi ovisiti o konekciji s internetom, može napraviti vlastiti primjerak Bootstrapa pa ga uključiti u HTML datoteku kao što je bio slučaj i sa CSS-om. [6]

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
```

Sl. 2.4. Kôd za ugrađivanje Bootstrapa na HTML stranici.

Na slici 2.5. prikazan je primjer kôda aplikacije gdje se koristi opcija navigacije iz Bootstrapa.

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="filmoteka.html">
        
      </a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        <li class="active"><a href="home.php">Početna</a></li>
        <li><a href="korisnici.php">Korisnici</a></li>
        <li><a href="filmovi.php">Filmovi</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Sl. 2.5. Primjer korištenja navigacije na vrhu stranice koristeći Bootstrap.

## 2.4. Javascript

Javascript je programski jezik koji služi za programiranje ponašanja web stranica. Vrlo je lagan za implementaciju jer je integriran s HTML-om. Činjenica je da je Javascript jedan od najpopularniji i najšire korištenih programskih jezika danas. [7]

Neke od koristi:

- Manipulacija HTML stranicama – Javascript ima mogućnosti da lako dodaje i briše bilo koje HTML oznake te tako mijenja izgled stranice na više različitih uređaja i zahtjeva.
- Notifikacija korisnika – Koristi se također i za obavješavanje posjetitelja stranice raznim dinamičkim skočnim prozorima (engl. *Pop-up ads*).
- Učitavanje Back-end podataka – Omogućava Ajax biblioteku koja poboljšava doživljaj gostima stranice tako što pomaže s učitavanjem podataka koji su back-end.
- Potvrđivanje klijenata – Vrlo je važno za provjeriti svaki korisnički unos prije slanja na server te je Javascript vrlo bitan prilikom provjere unosa na front-endu.

Na slici 2.6. vidimo primjer jednostavnog Javascript kôda korištenog u HTML-u koji ispisuje poznati „Hello World!“ na ekran. [8]

```
<html>
  <body>
    <script language = "javascript" type = "text/javascript">
      <!--
        document.write("Hello World!")
      //-->
    </script>
  </body>
</html>
```

Sl. 2.6. Primjer jednostavnog Javascript kôda.

Na slici 2.7. se vidi primjer kôda iz ove aplikacije koji se koristi prilikom gledanja filma, gdje korisnik upravlja je li film pokrenut ili ne. To je omogućeno uz *if-else* blok gdje provjeravamo stanje definirane varijable *myVideo*.

```
        </video>
        <br><br>
        <button onclick="playPause()">Play/Pause</button>
    </div>

<script>
var myVideo = document.getElementById("videol");

function playPause() {
    if (myVideo.paused)
        myVideo.play();
    else
        myVideo.pause();
}
</script>
```

Sl. 2.7. Primjer korištenja *playPause()* funkcije u mojoj aplikaciji.

## 2.5. PHP

PHP (engl. *Hypertext Preprocessor*) je skriptni jezik od strane servera i moćan alat koji služi da bi web stranice napravio dinamičnim i interaktivnim. Široko je korišten, besplatan i vrlo učinkovit. [9][10]

Neke od koristi:

- Može rukovati formama, na primjer prikupljanje podataka iz datoteka, spremanje podataka u datoteke, vratiti podatke korisniku i slično.
- Izvodi sistemske funkcije, na primjer iz datoteka sa sistema može kreirati, otvarati, čitati pisati i zatvarati iste.
- Služi za dodavanje, brisanje te uređivanje elemenata u bazi kojom upravljamo.
- Može se zabraniti korisnicima pristup određenim stranicama na stranici.
- Može šifrirati podatke.
- Pristup i postavka „kolačića“. [11]

Na slici 2.8. može se vidjeti primjer jednostavnog PHP kôda.

```
<html>

  <head>
    <title>Hello World</title>
  </head>

  <body>
    <?php echo "Hello, World!";?>
  </body>

</html>
```

Sl. 2.8. Primjer jednostavnog PHP kôda.

Na slici 2.9. se vidi PHP kôd iz aplikacije. Kôd je ugrađen u HTML te su zajedno kombinirani. Prije svakog pisanja ovakvog kôda uvijek je nužno napisati „<?php“, a na kraju „?>“. Prikazani kôd predstavlja *if-else* blok koji ispituje da li je korisnik prijavljen, te ukoliko jest prikazuje se njegovo korisničko ime te opcija za odjavu. U protivnom, vidi se tipka za prijavu te registraciju. Ključna riječ „echo“ služi za ispisivanje elemenata na ekran koji su pod php oznakama.

```
<?php if (!isLoggedIn()){
  echo "<ul class='nav navbar-nav navbar-right'>";
  echo "<li><a href='login.php'><span class='glyphicon glyphicon-log-in'></span> Prijava</a></li>";
  echo "<li><a href='login.php'> Registracija</a></li>";
  echo "</ul>";
}else{
  echo "<ul class='nav navbar-nav navbar-right'>";
  echo "<strong style='color:white;font-size:20px;'>";
  echo $_SESSION['user']['username'];
  echo "</strong><br/>";
  echo "<a href='index.php?logout=1' style='color:red;'><span class='glyphicon glyphicon-log-out'></span>&nbsp;logout</a>";
  echo "</ul>";
}
?>
```

Sl. 2.9. Primjer PHP kôda iz aplikacije.

### 3. OPIS FUNKCIONALNOSTI WEB APLIKACIJE

U ovom poglavlju opisane su sve funkcionalnosti aplikacije. Priloženi su kôdovi koji prikazuju kako se s tehničke strane došlo do određenih rješenja koja su bila potrebna za normalan rad stranice. Prilikom izrade svake web aplikacije, potreban je i rad s bazama podataka. Baza podataka ove aplikacije sastoji se od raznih tablica koje su korištene prilikom povezivanja podataka da bi se stranica učinila potpuno funkcionalnom. Na slici 3.1. može se vidjeti primjer spajanja na bazu.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

**Sl. 3.1.** Spajanje na bazu PHP-om.

Putem PHP-a spaja se na određeni server. Potrebno je deklarirati varijable koje u PHP-u započinju sa znakom „\$“. Nakon što se ostvari konekcija, dalje se provjerava mogućnost pojave pogreške.



Nadalje, prikazan je kôd napisan u PHP-u prilikom registracije korisnika u videoteku. Dio kôda je prikazan na slici 3.2.

```
function register(){
    global $db, $errors, $username, $email;

    $username = e($_POST['username']);
    $email = e($_POST['email']);
    $password_1 = e($_POST['password_1']);
    $password_2 = e($_POST['password_2']);

    if (empty($username)) {
        array_push($errors, "Username is required");
    }
    if (empty($email)) {
        array_push($errors, "Email is required");
    }
    if (empty($password_1)) {
        array_push($errors, "Password is required");
    }
    if ($password_1 != $password_2) {
        array_push($errors, "The two passwords do not match");
    }
    if (count($errors) == 0) {

        if (isset($_POST['user_type'])) {
            $user_type = e($_POST['user_type']);
            $query = "INSERT INTO users (username, email, user_type, password)
                VALUES('$username', '$email', '$user_type', '$password')";
            mysqli_query($db, $query);
            $_SESSION['success'] = "New user successfully created!!";
            header('location: home.php');
        }else{
            $query = "INSERT INTO users (username, email, user_type, password)
                VALUES('$username', '$email', 'user', '$password')";
            mysqli_query($db, $query);
        }
    }
}
```

Sl. 3.2. Kôd za registraciju korisnika videoteke.

Kao što se može vidjeti u kôdu, na početku varijablama dodajemo vrijednosti koje su se putem „POST“ metode poslale u formi. Nakon tog, izvršava se potrebna provjera koja nam kaže je li korisnik ostavio koje polje prazno te šalje poruku upozorenja ukoliko jest. Ako je sve prošlo bez greške, korisnici, odnosno administratori se upisuju u bazu podataka SQL naredbom „INSERT INTO“ te se ispisuju poruke ovisne o slučaju.

Na slici 3.3. prikazan je kôd vrlo sličan onom za registraciju korisnika aplikacije. A radi se o kôdu za prijavu.

```
function login(){
    global $db, $username, $errors;

    $username = e($_POST['username']);
    $password = e($_POST['password']);

    if (empty($username)) {
        array_push($errors, "Username is required");
    }
    if (empty($password)) {
        array_push($errors, "Password is required");
    }

    if (count($errors) == 0) {
        $password = md5($password);

        $query = "SELECT * FROM users WHERE username='$username' AND password='$password' LIMIT 1";
        $results = mysqli_query($db, $query);

        if (mysqli_num_rows($results) == 1) {
            $logged_in_user = mysqli_fetch_assoc($results);
            if ($logged_in_user['user_type'] == 'admin') {

                $_SESSION['user'] = $logged_in_user;
                $_SESSION['success'] = "You are now logged in";
                header('location: home.php');
            }else{
                $_SESSION['user'] = $logged_in_user;
                $_SESSION['success'] = "You are now logged in";

                header('location: index.php');
            }
        }else {
            array_push($errors, "Wrong username/password combination");
        }
    }
}
```

Sl. 3.3. Kôd za prijavu korisnika videoteke.

Također se na početku postavljaju varijable koje su se dobile, ovaj put iz login forme ali isto putem „POST“ metode. Kada se izvrše sve provjere, u bazi se putem naredbe „SELECT“ biraju sva polja gdje se podaci podudaraju s unesenim, te ukoliko se dogodi poklapanje, preusmjerava se korisnika ovisno o njegovom tipu.

Na slici 3.4. je prikazan vrlo bitan dio kôda pisan u PHP-u koji je često korišten tokom izrade aplikacije.

```
function isLoggedIn()
{
    if (isset($_SESSION['user'])) {
        return true;
    }else{
        return false;
    }
}

if (isset($_GET['logout'])) {
    session_destroy();
    unset($_SESSION['user']);
    header("location: index.php");
}
```

Sl. 3.4. Pomoćni kôdovi za provjeru logiranja te za odjavu.

Funkcijom *isLoggedIn()* provjerava da li je tokom posjete stranici korisnik prijavljen ili ne. Ako postoji sesija određenog korisnika, funkcija će vratiti istinitu vrijednost i suprotno. Ova funkcija služi u mnogo dijelova kôda gdje se npr. određena tipka, ili određeni element treba prikazati samo onome tko je prijavljen. U drugom dijelu slike vidimo *if* blok kôda gdje se uništava sesija ukoliko se putem „*GET*“ metode dobije znak da je korisnik kliknuo na odjavu.

Na slici 3.5. prikazan je jednostavan kôd pomoću kojeg se manipulira bazom i na taj način se briše korisnika.

```
<?php
session_start();
include_once("functions.php");

if(!isset($_GET['uid'])){
    header("Location: korisnici.php");
} else{
    $uid = $_GET['uid'];
    $sql = "DELETE FROM users WHERE id=$uid";
    mysqli_query($db, $sql);
    header("Location: korisnici.php");
}
?>
```

Sl. 3.5. Kôd za brisanje korisnika iz baze.

Ovaj dio kôda se naziva i takozvanom više namjenskom stranicom pomoću kontrolne riječi. U ovome slučaju kontrolna riječ predstavlja „uid“ koji označava korisnika koji se veže sa svojim id-em. (npr. del\_user.php?uid=\$id, sve poslije upitnika predstavlja kontrolnu riječ). Nakon što je utvrđeno o kojem je korisniku riječ, naredbom „DELETE“ briše se korisnik iz baze. Na sličan način izvršava se i brisanje filmova iz baze.

Na slici 3.6. prikazan je način na koji se upisuju ocjene u bazu podataka nakon što korisnik klikne na određenu tipku.

```
<?php
include('functions.php');

$user_id = $_GET['userid'];
$film_id = $_GET['filmid'];
$rating = $_GET['rating'];

$sql = "SELECT * FROM ocjene WHERE IDfilma = '$film_id' AND IDosobe = '$user_id'";
$res = mysqli_query($db,$sql);

if(mysqli_num_rows($res) > 0) {
    $sql = "UPDATE ocjene SET ocjena = '$rating' WHERE IDosobe = '$user_id' AND IDfilma = '$film_id'";
    mysqli_query($db,$sql);
}
else{
    $sql = "INSERT INTO ocjene (ocjena, IDosobe, IDfilma) VALUES('$rating','$user_id','$film_id')";
    mysqli_query($db,$sql);
}

echo "success";
die();
?>
```

Sl. 3.6. Kôd za dodavanje ocjene u bazu.

Na početku varijablama se dodjeljuju vrijednosti koje su šalju metodom „GET“ u Javascript kôdu. Nakon tog pravi se upit SQL-u putem kojeg se odabire ID filma i osobe koja odgovara zadanim varijablama. Ako je korisnik već ocijenio film sa tim ID-em, ocjena će se samo ažurirati naredbom „UPDATE“, dok će se u suprotnom ocjena unijeti prvi put u tablicu „ocjene“.

Svaka web aplikacija na internetu koristi određeni način za dobivanje podataka iz baze kao i unos u nju, a u ovom slučaju to je *phpMyAdmin*. Na ovaj način mogu se vršiti svakakve manipulacije bazom. Na slici 3.7. može se vidjeti prikaz baze podataka koja služi za spremanje podataka aplikacije za vođenje online videoteke.

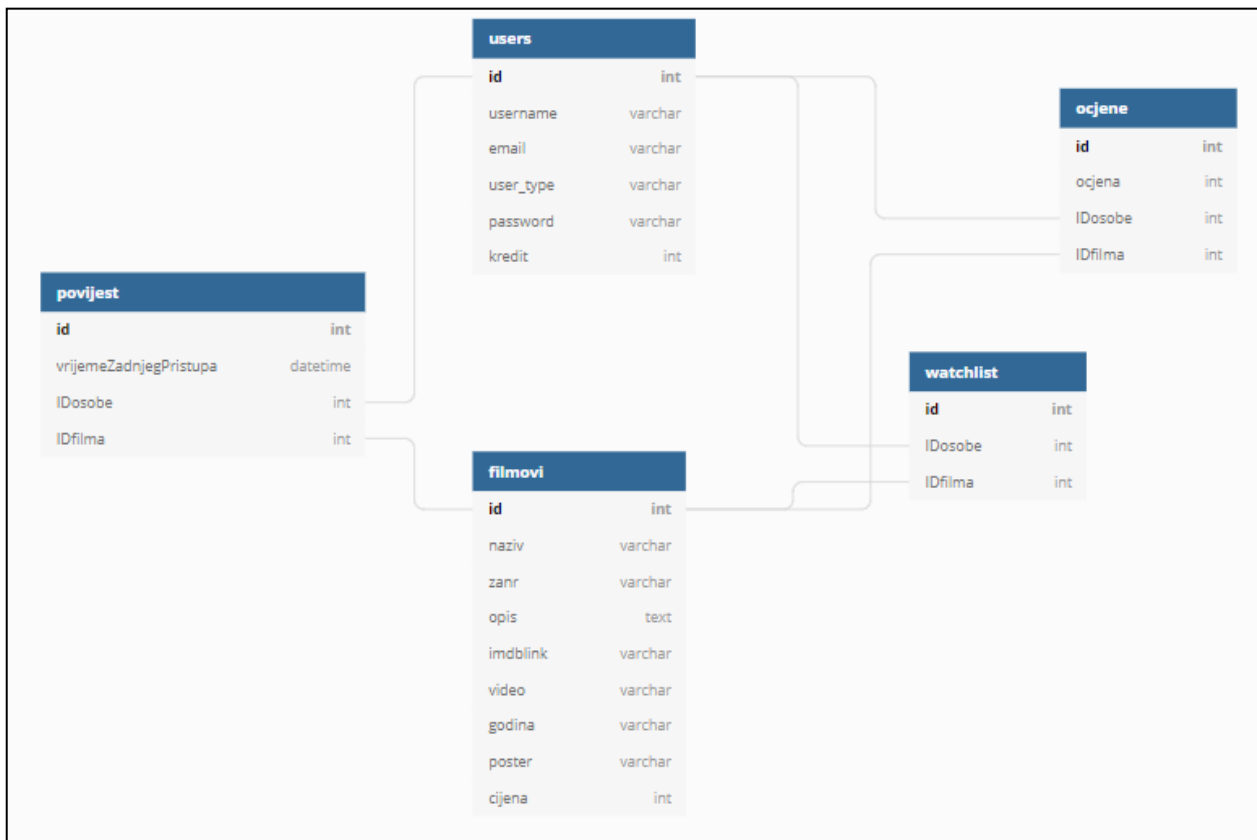
Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> filmovi	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> ocjene	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_croatian_ci	16 KiB	-
<input type="checkbox"/> povijest	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8_croatian_ci	16 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> watchlist	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	48 KiB	-
5 tables	Sum	39	MyISAM	utf8mb4_general_ci	112 KiB	0 B

**Sl. 3.7.** Baza podataka web aplikacije za vođenje online videoteke.

Kao što se može vidjeti na slici, za izradu ove aplikacije kreirane su tablice filmovi, ocjene, povijest, korisnici (engl. *users*) i lista gledanja. Tablica filmove sadrži sve informacije potrebne za određeni film kojeg administrator doda, a to su naziv, žanr, godina izdavanja, poster, opis, veza (engl. *link*), video, cijena te naravno id filma kojeg se kasnije može koristiti kod povezivanja više baza podataka. Tablica ocjene sadrži id korisnika koji je dao ocjenu, id filma koji je ocijenjen, te naravno ocjenu.

Tablica povijest služi za informacije o tome kada je korisnik zadnji put posudio neki film te stavio ga u listu gledanja, a sadrži id korisnika, id filma te vrijeme zadnjeg pristupa. Tablica korisnici služi u početku kod prijave te registracije. Kada se korisnik registrira upisuje neke od podataka koji se onda unose u ovu tablicu, a neki su korisničko ime, e-mail adresa i lozinka. Ovdje se također nalazi i vrlo bitan podatak kojem samo administrator može pristupiti, a to je tip korisnika koji je određeni posjetitelj videoteke. Na kraju tu je i tablica lista gledanja koja se sastoji od podataka kao što su id korisnika te filma.

Na slici 3.8. prikazan je dijagram koji prikazuje kako su tablice u bazi podataka povezane.



Sl. 3.8. Dijagram baze podataka.

Na slici 3.9. prikazan je Javascript kôd koji služi za dodavanje filma u listu gledanja.

```
function addToWatchlist(){
    if(document.getElementById("button1").innerHTML == "Ukloni"){
        removeFromWatchlist();
        return;
    }
    var osobaId = <?php echo $id_osobe ?>;
    var filmId = <?php echo $id ?>;

    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            if(this.responseText == "success"){
                alert("Uspješno dodano.");
                document.getElementById("button1").innerHTML = "Ukloni";
                location.reload();
            }
            else if(this.responseText == "Not enough funds."){
                alert("Nemate dovoljno novca na računu.");
            }
            else{
                alert("Došlo je do greške. Pokušajte ponovno.");
            }
        }
    };
    xmlhttp.open("GET", "addtowatchlist.php?userid=" + osobaId + "&filmid=" + filmId , true);
    xmlhttp.send();
}
```

Sl. 3.9. Kôd za dodavanje filma u listu gledanja.

Funkcija prikazana u kôdu poziva se ukoliko korisnik klikne na tipku na kojoj piše dodaj. Ukoliko na tipki piše ukloni (što znači da je film već prije dodan u listu) poziva se funkcija *removeFromWatchlist()* koja je definirana ispod. Prvo se definiraju varijable gdje se putem baze dobivaju vrijednosti prijavljene osobe, odnosno filma na čijem smo sučelju. Nakon tog kroz određene Javascript funkcije ispisujemo korisniku poruke na osnovu ishoda.

Na slici 3.10. prikazan je Javascript kôd koji nam služi za ocjenjivanje filmova.

```
function rate(){
    selector = document.getElementById("ratingSelect");
    rating = selector.options[selector.selectedIndex].value;
    var osobaId = <?php echo $id_osobe ?>;
    var filmId = <?php echo $id ?>;

    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            if(this.responseText == "success"){
                alert("Ocijenili ste film s: " + rating);
            }
            else{
                alert("Došlo je do greške. Pokušajte ponovno.");
            }
        }
    };
    xmlhttp.open("GET", "ratemovie.php?userid=" + osobaId + "&filmid=" + filmId + "&rating=" + rating , true);
    xmlhttp.send();
}
```

**Sl. 3.10.** Kôd za ocjenjivanje filma.

Na slici 3.10. prikazan je sličan kôd kao sa slike 3.9. Putem određenih Javascript funkcija korisniku se ispisuju poruke ovisne o njihovim odabirima. Nakon tih funkcija, šalju se određene informacije drugoj datoteci gdje se izvršava PHP kôd te manipulira bazom podataka.

## 4. KORIŠTENJE WEB APLIKACIJE

U ovom poglavlju objašnjen je rad aplikacije. Prvo je prikazana aplikacija iz perspektive administratora, a nakon toga iz perspektive korisnika videoteke. Priloženim slikama opisan je rad stranice nakon klika na određene tipke, te su dane upute kako doći do određenih dijelova stranice.

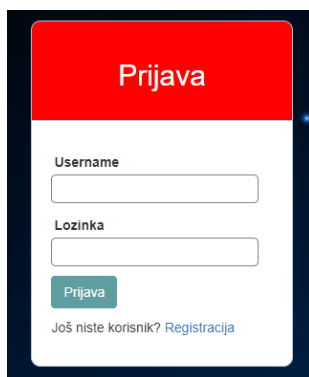
### 4.1. Upute za korištenje aplikacije administratoru

Jedan od subjekata pri korištenju ove aplikacije je administrator. On posjeduje određene mogućnosti i opcije koje ne posjeduju ostali korisnici. U posebnom sučelju koje je stvoreno samo za administratora, on se prijavljuje te ima dozvoljene određene ovlasti. Administrator ima kompletan uvid u podatke o svim korisnicima online videoteke. Stvoreno je posebno sučelje koje vidi samo on, te je za korisnika nedostupno i nevidljivo.

Kao još jednu posebnu opciju administrator za razliku od ostalih korisnika ima mogućnost unošenja novih filmova u bazu, koji nisu prije postojali. Još jedna vrlo bitna mogućnost je brisanje filmova iz te baze. U nastavku je prikazano sučelje od administratora, te kako se isto razlikuje od korisničkog. Također je prikazan i kôd koji je jedinstven samo za njegovu stranu.

#### 4.1.1. Prijava administratora

Administrator i korisnik imaju isti login obrazac. U *mysql* bazi podataka u kojoj se nalaze svi subjekti ove stranice postoji kolona smisleno nazvana „*user\_type*“. Prilikom prijave provjerava se da li je prijavljena osoba administrator ili korisnik. Ako je administrator otvara mu se stranica *home.php* koju samo korisnik s tipom '*admin*' može vidjeti. Jedino administratori mogu stvarati nove administratore. Na slici 4.1. prikazana je login forma.

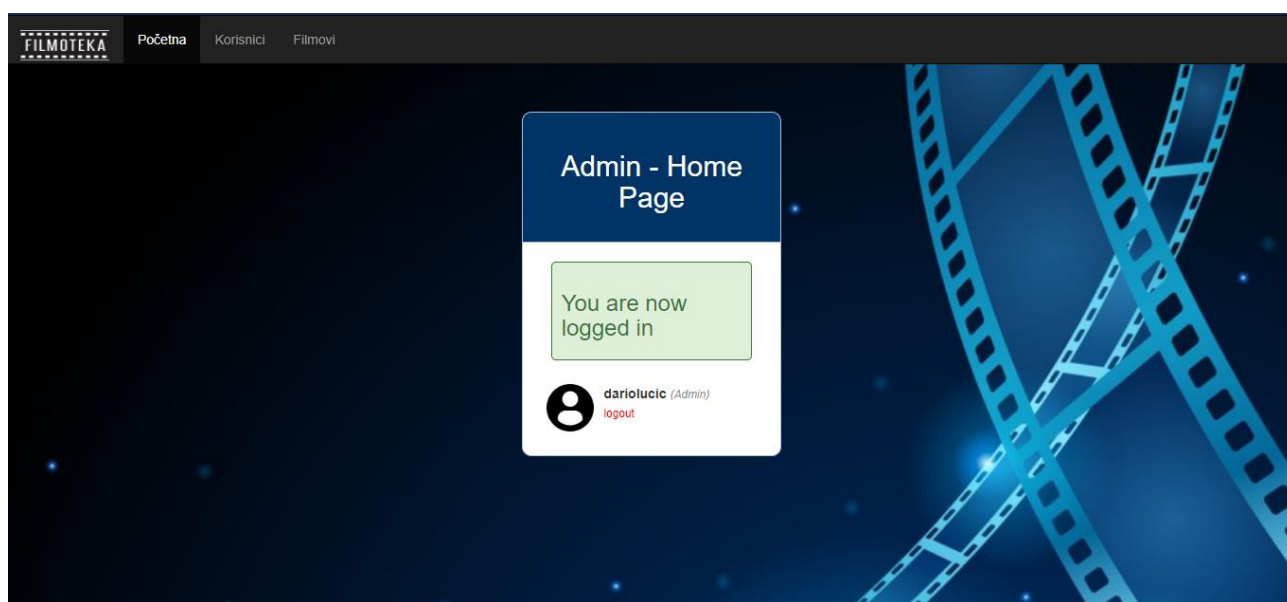


Sl. 4.1. Login forma za prijavu administratora web stranice.



### 4.1.2. Stranica administratora

Nakon prijave zaključeno je kako je „*user\_type*“ iz baze podataka osobe koja se prijavila jednak ključnoj riječi '*admin*', te se stranica automatski preusmjerava na administratorovu glavnu stranicu. Stranica nakon prijave administratora prikazana je na slici 4.2.

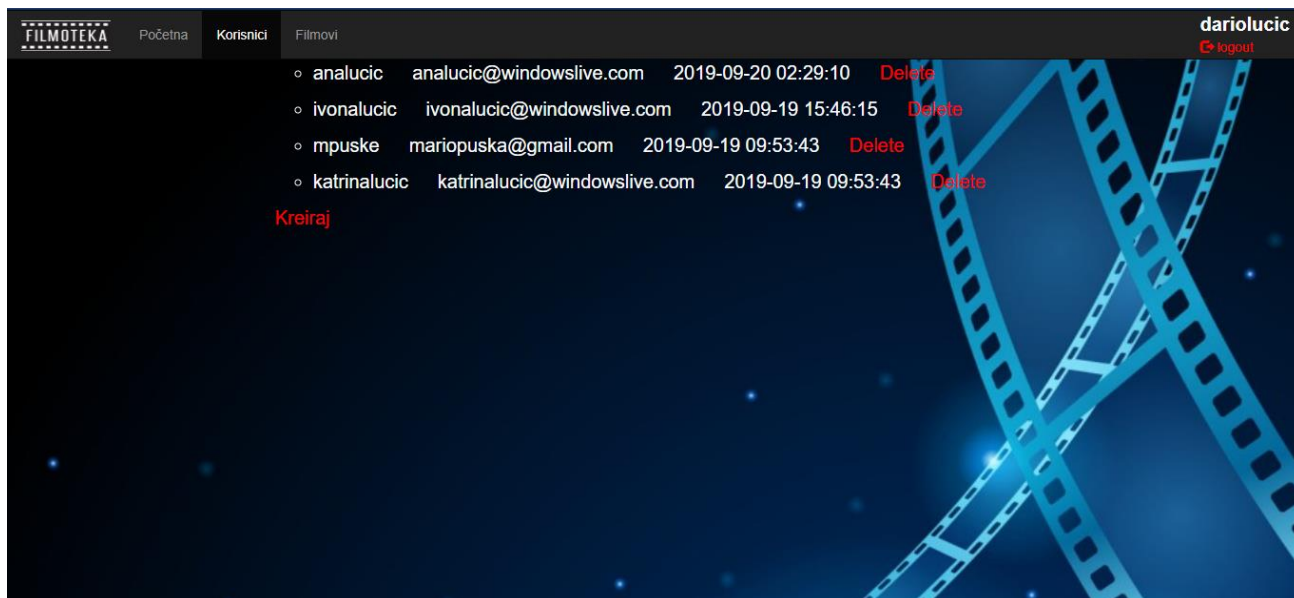


Sl. 4.2. Glavna stranica administratora.

Na slici je prikazano kako se otvara novo sučelje prilagođeno samo potrebama administratora. Administrator u izborniku ima ponuđene opcije gdje može vidjeti sve filmove i korisnike videoteke, te su mu omogućene dodatne opcije za dodavanje i brisanje istih iz baze. O tome nešto više u daljnjem tekstu.

### 4.1.3. Izbornik-korisnici

Kada je već osigurano i provjereno da je subjekt koji se prijavio administrator, otvara se sesija koja će omogućiti da samo on vidi zadane stranice. Nakon što se uspješno prijavio, administrator može otići na stranicu gdje vidi sve korisnike tako što će u izborniku odabrati '*Korisnici*'. Kao što je prikazano na slici 4.3., kada administrator posjeti tu stranicu, moći će vidjeti sve korisnike koji se nalaze u bazi podataka te koji su registrirani u videoteci. Uz svakog prikazanog korisnika ima opciju da istog izbríše iz baze.



Sl. 4.3. Stranica gdje administrator vidi sve korisnike videoteke.

Administratoru je također ponuđena opcija da kreira nove korisnike, a uz tu opciju može kreirati i administratore kao što je objašnjeno u daljnjem tekstu. U gornjem desnom kutu nalazi se ime administratora, kao i opcija za odjavu.

#### 4.1.4. Dodavanje novog korisnika

Nakon što je već prijavljeni administrator odabrao opciju 'Kreiraj', otvara mu se novi prozor gdje ima opciju dodavanja novih korisnika u bazu podataka. Kao što je prikazano na slici 4.4. omogućena mu je forma gdje sam može upisivati sve podatke koji su potrebni da bi se stvorio novi subjekt. Također ima zadanu opciju koja omogućava odabiranje tipa novostvorenog subjekta, te može stvoriti i sebi ekvivalentnog administratora koji će imati iste mogućnosti kao i on nakon prijave.

Admin - kreiraj korisnika

Username

Email

Tip korisnika

Lozinka

Potvrđi lozinku

+ Kreiraj korisnika

Sl. 4.4. Forma za unos novih korisnika u videoteku.

#### 4.1.5. Izbornik-filmovi

Kada je već osigurano i provjereno da je subjekt koji se prijavio administrator, otvara se sesija koja će omogućiti da samo on vidi zadane stranice. Nakon što se uspješno prijavio administrator može otići na stranicu gdje vidi sve korisnike tako što će u izborniku odabrati „*Filmovi*“.

Kada administrator posjeti tu stranicu može vidjeti sve filmove koji se nalaze u bazi podataka. Uz naziv i opis filma on može vidjeti i sve ostale atribute koji imaju filmovi u bazi, te ima razne opcije. Uz sve navedene opcije koje ima, može brisati filmove iz baze, uređivati ih ako je došlo do nekih promjena ili pogrešaka, te dodavati nove filmove. Na slici 4.5. prikazano je sučelje koje se otvara administratoru nakon što je u izborniku odabrao '*Filmovi*'.



Sl. 4.5. Stranica gdje administrator vidi sve filmove u videoteci.

## 4.2. Upute korisnicima za korištenje aplikacije

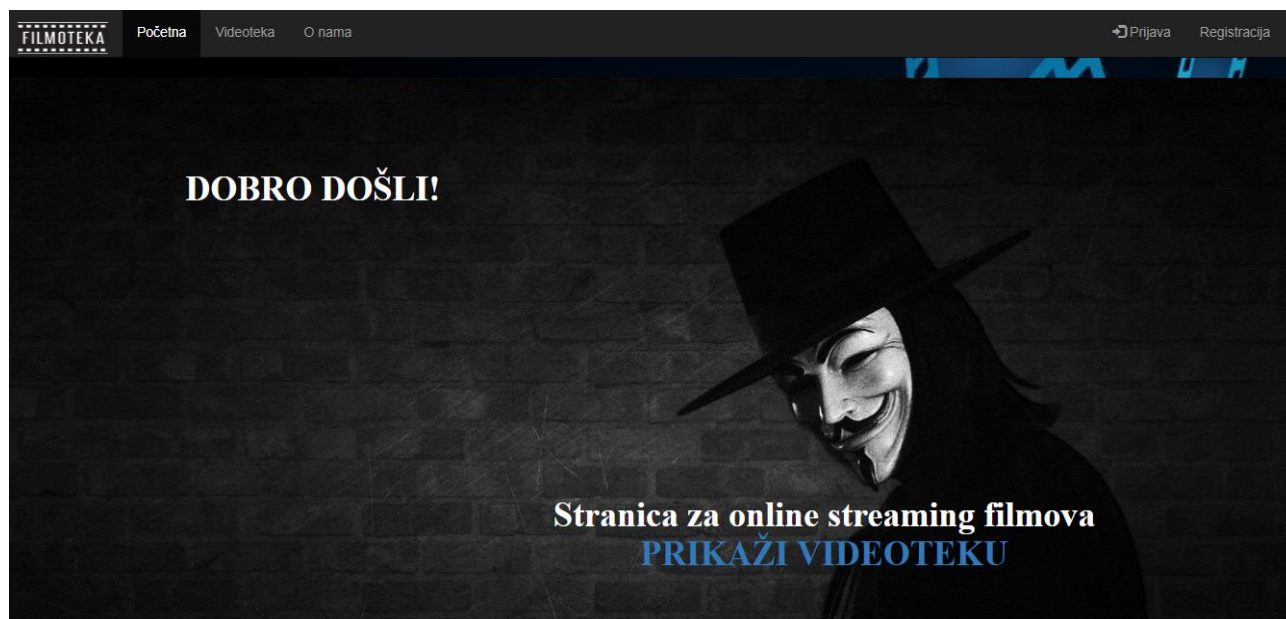
U ovom poglavlju opisan je izgled aplikacije iz drugog pogleda, iz perspektive korisnika. Bit će prikazano korisničko sučelje, te kako se ono razlikuje od svega čime administrator rukuje. Korisnicima su ponuđene različite opcije, te je ovdje opisano i prikazano kako to izgleda, te kako da korisnik rukuje istim.

Nakon što se prijave, korisnici mogu upravljati svojom listom filmova gdje se nalazi popis onih koje je korisnik posudio, te su se dodali na tu listu. Korisnik također može vidjeti status svoje članarine, te ima uvid u povijest svojih posudbi. Svaki film ima određenu cijenu te korisnik mora imati odabran neki od načina plaćanja (tipa kreditna kartica) gdje će mu se skidati novac s računa.

Kao dodatnu opciju, korisnik može svaki film ocijeniti te će tako pridonijeti ukupnoj ocjeni te pomoću ostalim korisnicima pri izboru za njihovo gledanje. Kako korisnici mogu dodavati filmove u svoju listu, isto tako može ih i uklanjati s nje. U nastavku je prikazano kako izgleda korisničko sučelje te kôd u kojem se ovaj dio aplikacije razlikuje od stranice administratora.

## 4.2.1. Početna stranica i prijava

Na slici 4.6. prikazan je izgled početne stranice aplikacije.



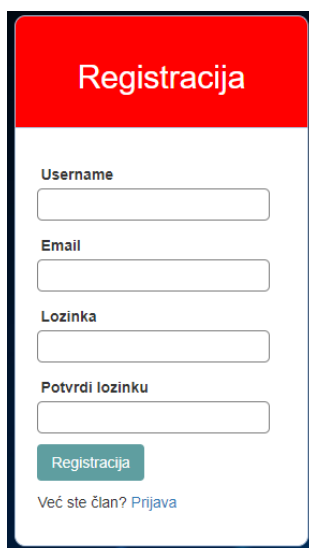
Sl. 4.6. Početna stranica.

Na početnoj stranici je prikazano da je korisnicima ponuđena opcija da posjete videoteku. U gornjem desnom kutu vidimo opcije prijave i registracije, koje su vidljive samo onda kada nitko nije prijavljen. Kod prijave se koristi isti login obrazac kao i kod prijave administratora. Razlika je ovdje što će se prijavljivati korisnici te će im biti drugačije sučelje nakon prijave.

Na slici 4.7. prikazan je login forma.

Sl. 4.7. Login forma za prijavu korisnika web stranice.

Kada je o registraciji riječ, korisnici koji nisu dosad prijavljeni imaju malo drugačiji obrazac koji moraju ispuniti te se tako uneseni podaci uz sve ispravne unose kreiraju u bazi podataka gdje se nalaze korisnici, obrazac je prikazan na slici 4.8.

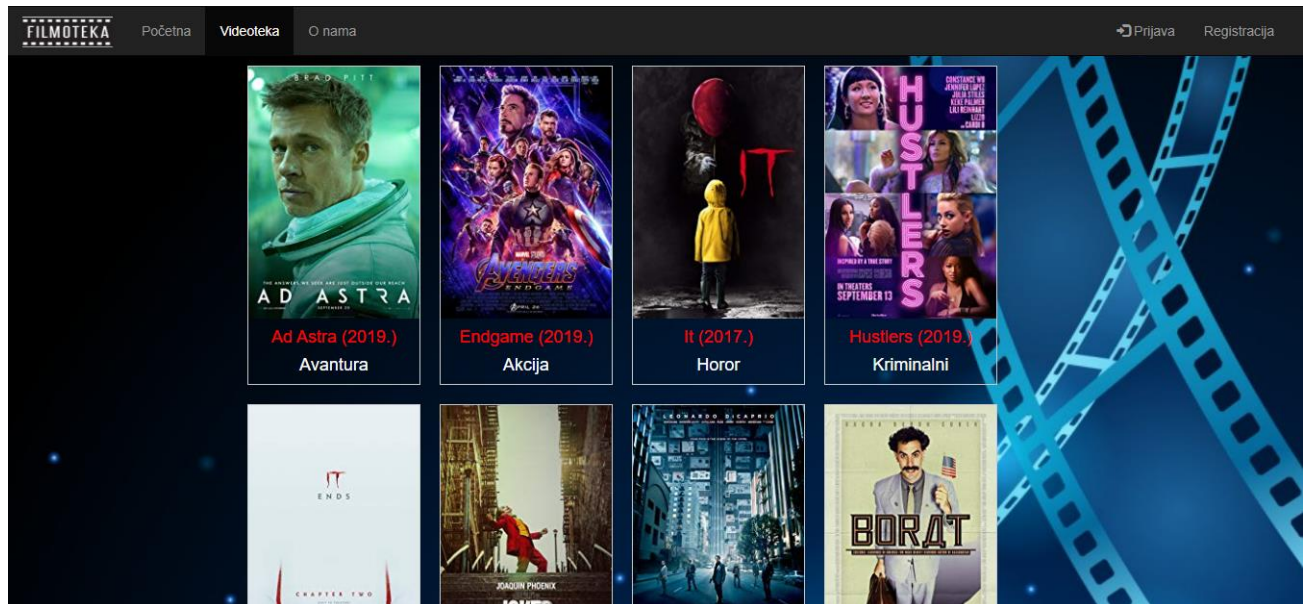


The image shows a registration form with a red header containing the word "Registracija" in white. Below the header, there are four input fields: "Username", "Email", "Lozinka", and "Potvrdi lozinku". Each field is a simple white rectangle with a thin border. Below the "Potvrdi lozinku" field is a green button with the text "Registracija" in white. At the bottom of the form, there is a link that says "Već ste član? [Prijava](#)".

**Sl. 4.8.** Forma za registraciju korisnika web stranice.

#### 4.2.2. Videoteka

Svaki korisnik i prije prijave na stranicu može u izborniku odabrati „Videoteka“, ili preko glavne stranice odabrati „*PRIKAŽI VIDEOTEKU*“. Nakon što odabere jedan od dva načina korisnik može vidjeti prikazan popis svih filmova koji se nalaze u bazi. Za svaki film prikazan je naziv i godina izdavanja, žanr te poster. Filmovi su poslagani u obliku galerije tako da se korisnik može lakše snaći te da ima bolji pregled. Videoteka je prikazana na slici 4.9.

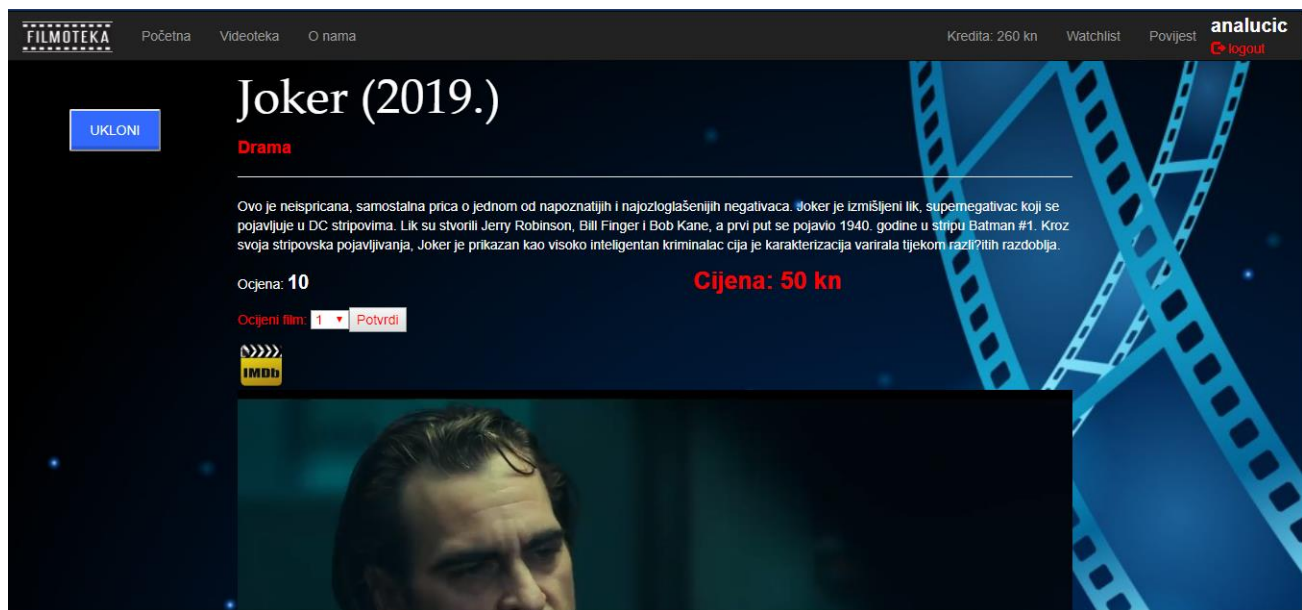


Sl. 4.9. Prikaz videoteke.

### 4.2.3. Gledanje filma

Nakon što korisnik klikne na određeni film iz videoteke, otvara mu se novo sučelje gdje može birati različite opcije na osnovu toga da li je prijavljen ili ne. Ukoliko je na stranici samo regularni posjetitelj, on neće moći gledati film te neće imati mogućnost ocjenjivanja filma, ali će moći vidjeti prosječnu ocjenu korisnika za film.

Ukoliko je korisnik prijavljen, također mu se otvara sučelje odabranog filma ali s mnogo više opcija. Na vrhu stranice može vidjeti sve bitne informacije o filmu: naziv, godinu izdavanja, žanr, opis filma, te ponuđena opcija za gledanje filma. Korisniku je ponuđena i opcija da klikne na dodavanje filma u listu gledanja, koja se onda može posjetiti u gornjem desnom uglu pokraj imena korisnika. Ukoliko je korisnik već dodao film u listu gledanja, prikazati će mu se tipka za uklanjanje filma iz iste. Na slici 4.10. prikazano je sučelje prilikom gledanja filma.



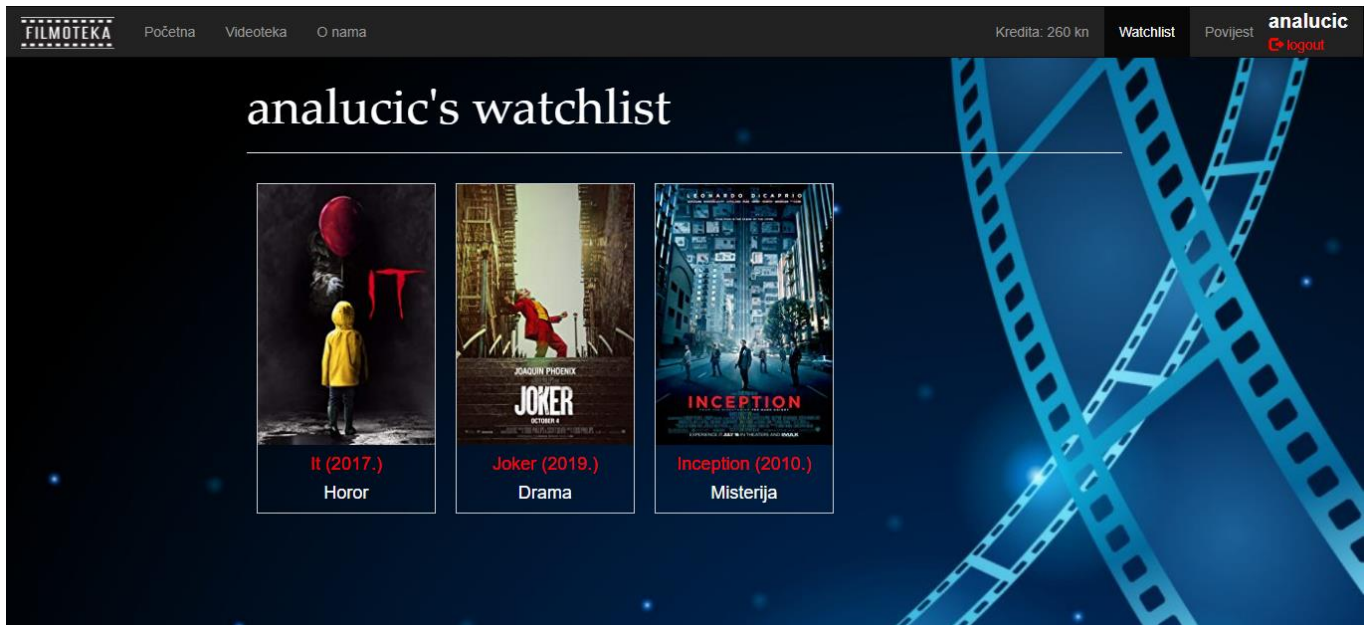
Sl. 4.10. Prikaz gledanja filma.

#### 4.2.4. Lista gledanja

Ukoliko je korisnik aplikacije prijavljen, pojavljuje mu se dodatna opcija u izborniku, a to je lista gledanja. Prilikom ulaska u svaki film koji postoji u videoteci, bit će mu ponuđena opcija da doda određeni film u listu.

Nakon što korisnik u izborniku odabere „Watchlist“, bit će mu prikazan popis svih filmova koje je stavio u nju, te će moći lakše pristupiti filmovima, bez potrebe da pregledaju cijelu videoteku svaki put. Na slici 4.11. prikazana je lista gledanja.

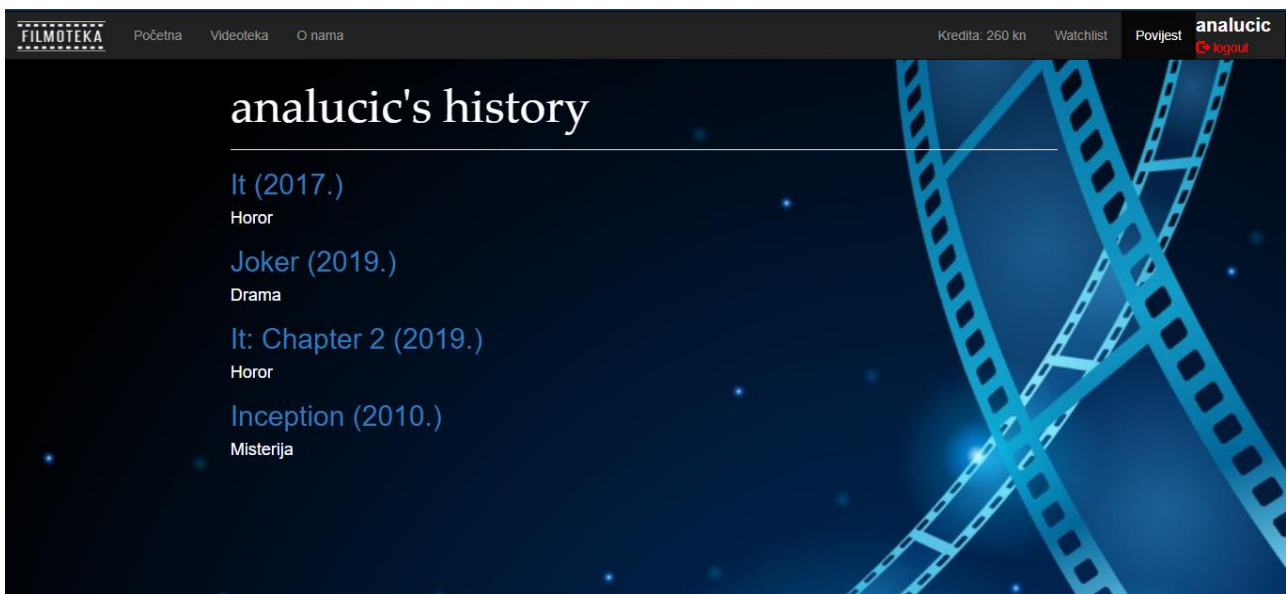




Sl. 4.11. Lista gledanja.

#### 4.2.5. Povijest dodavanja u listu

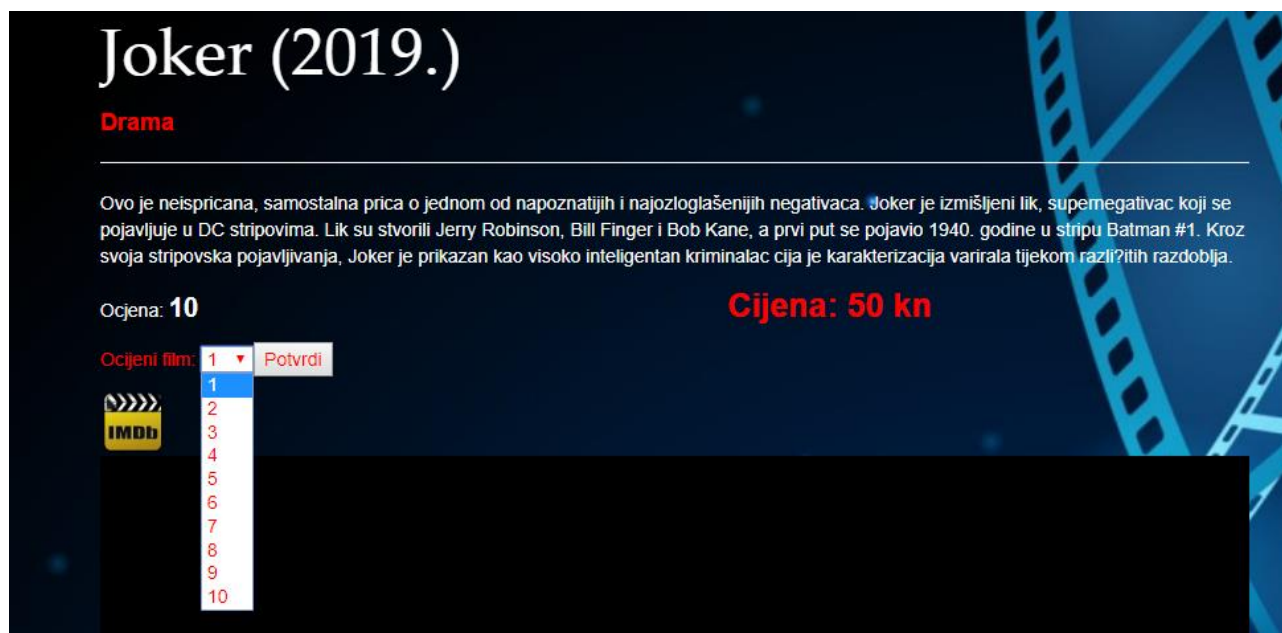
Kada se korisnik prijavi, u izborniku mu se pojavljuje nova opcija, povijest njegovog posuđivanja. Iako je korisnik uklonio određeni film s liste gledanja, u povijesti će mu biti zapisani filmovi koje je gledao. Filmovi mu se redaju redosljedom kako ih je dodavao, odnosno zadnji prikazani će biti onaj kojega je zapravo zadnjeg i stavio „na gledanje“. Na slici 4.12. prikazano je sučelje povijesti, odnosno popis filmova koje je korisnik posudio do sada.



Sl. 4.12. Povijest korisnikovog posuđivanja.

## 4.2.6. Ocjenjivanje filma

Iako korisnik nije prijavljen, može vidjeti trenutnu ocjenu filma. Ta ocjena zapravo predstavlja prosjek ocjena svih korisnika videoteke. Ako je korisnik prijavljen, njemu je ponuđeno ocjenjivanje. Korisnici mogu i promijeniti svoju odluku te ocijeniti film ponovno. Na slici 4.13. prikazan je način na koji korisnik daje ocjenu filmu.

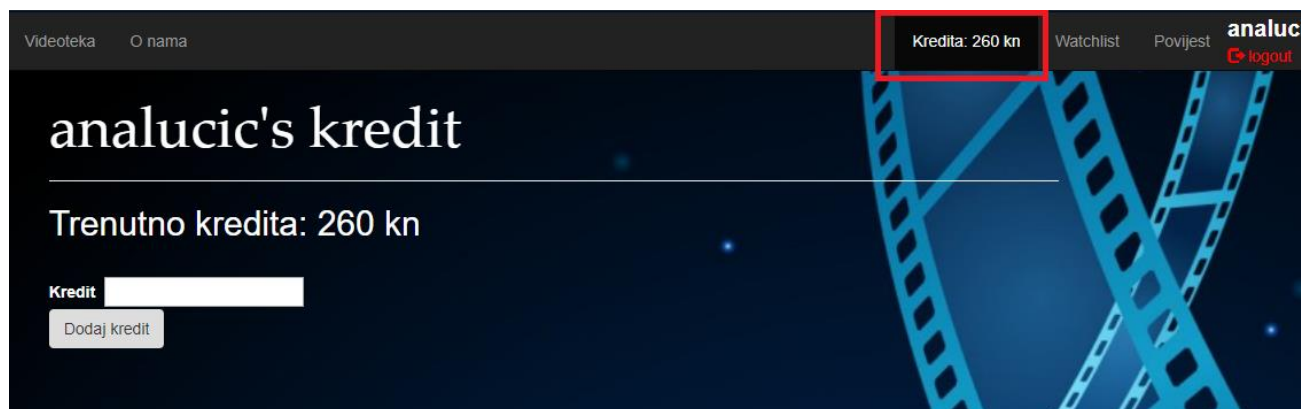


Sl. 4.13. Ocjenjivanje filma.

## 4.2.7. Kredit i plaćanje filma

Svaki korisnik koji je prijavljen ima svoj kredit izražen u kunama te njegovo stanje. Nakon prijave, u izborniku mu se pojavljuje prikaz gdje vidi koliko ima kredita na računu. Prilikom registracije, svaki korisnik dobija 100 kn kredita te na taj način može dodavati filmove u listu gledanja.

Svaki film ima svoju cijenu, filmovi koji su noviji i popularniji imaju veću cijenu od nekih stariji ili manje kvalitetni. Ukoliko korisnik doda film u svoju listu, skida mu se novac s računa. Ako korisnik nema dovoljnu količinu novca za dodavanje filma, dolazi mu obavijest kako mu nedostaje sredstava. Kada korisnik u izborniku odabere kredit, otvara mu se sučelje koje je zamišljeno kao simulacija gdje korisnik može dodati novac na svoj račun. To je prikazano na slici 4.14.



Sl. 4.14. Korisničko dodavanje kredita na račun.

## 5. ZAKLJUČAK

Cilj ovog rada bio je izgraditi web aplikaciju za vođenje online videoteke. Tehnologije koje su korištene za izradu aplikacije su HTML, CSS, Javascript, Bootstrap i PHP, dok je za bazu podataka korišten phpMyAdmin. Aplikacija je podijeljena na dva dijela, a to su aplikacija iz perspektive korisnika te aplikacija iz perspektive administratora. I korisnici i administratori imaju posebne načine korištenja i funkcionalnosti koje su posebne samo za njihovu stranu.

Kako je online gledanje filmova sve popularnije, ovom aplikacijom se mogu koristiti svi, a administratorova vrlo bitna uloga je da obogaćuje videoteku sa novim sadržajem kako bi se održao interes korisnika. Korisnicima je uz gledanje filma osigurano još i dodavanje filmova u svoju listu gledanja kako bi im se omogućilo lakše korištenje aplikacijom, te ocjenjivanje filmova kako bi pomogli i drugim korisnicima u odabiru filma za gledanje.

Ova aplikacija je vrlo korisna što se tiče korisničkih potreba za gledanje filmova, ali naravno da ima još mnogo prostora za napredak, te za poboljšanje određenih aspekata kao što su izgled te određene funkcionalnosti koje bi korisniku osigurale još bolji doživljaj prilikom gledanja.

## LITERATURA

- [1] Skupina autora, HTML Tutorial [online], TutorialsPoint, dostupno na: <https://www.tutorialspoint.com/html/index.htm> [10.09.2019.]
- [2] Skupina autora, HTML Tutorial [online], w3schools, dostupno na: <https://www.w3schools.com/html/> [10.09.2019.]
- [3] Skupina autora, CSS Tutorial [online], w3schools, dostupno na: <https://www.w3schools.com/css/> [10.09.2019.]
- [4] Skupina autora, CSS Tutorial [online], TutorialsPoint, dostupno na: <https://www.tutorialspoint.com/css/index.htm> [10.09.2019.]
- [5] Skupina autora, Bootstrap Tutorial [online], w3schools, dostupno na: <https://www.w3schools.com/bootstrap/> [10.09.2019.]
- [6] Skupina autora, Detailed Bootstrap Tutorial for Beginners [online], WebsiteSetup, dostupno na: <https://websitesetup.org/bootstrap-tutorial-for-beginners/> [10.09.2019.]
- [7] Skupina autora, Javascript Tutorial [online], w3schools, dostupno na: <https://www.w3schools.com/js/> [10.09.2019.]
- [8] Skupina autora, Javascript Tutorial [online], TutorialsPoint, dostupno na: <https://www.tutorialspoint.com/javascript/index.htm> [10.09.2019.]
- [9] Skupina autora, PHP Tutorial [online], w3schools, dostupno na: <https://www.w3schools.com/php/> [10.09.2019.]
- [10] Skupina autora, Uvod u PHP [online], php.com.hr, dostupno na: <https://php.com.hr/77> [10.09.2019.]
- [11] Skupina autora, PHP Tutorial [online], TutorialsPoint, dostupno na: <https://www.tutorialspoint.com/php/index.htm> [10.09.2019.]

## **SAŽETAK**

### **Naslov: Web aplikacija za vođenje online videoteke**

Zadatak ovog rada je kreiranje web aplikacije za vođenje online videoteke. U aplikaciji postoje administrator te korisnici. Svrha aplikacije je da je posjetitelji koriste za gledanja filmova.

Administrator web aplikacije ima uvid u sve članove videoteke. Korisnici imaju uvid u cjenik ponuđenih filmova te povijest svog gledanja. Administrator ima mogućnosti unošenja filmova u bazu, brisanje iz baze te dodatne opcije. Kada se korisnik prijavi, također ima mogućnost brisanja filmova iz košarice, te može posuđivati nove filmove kao i ocjenjivati ih.

**Ključne riječi:** baza podataka, bootstrap, filmovi, videoteka, web stranica

## **ABSTRACT**

### **Title: Web application for managing online movie library**

The premise of this paper is creating web application for managing an online video library. Subjects in this app are administrators and users. Purpose of the application is for visitors to use it for watching movies.

Administrator has insight into all users of the application. Users have insight into catalog of movies as well as history of watching. Administrator has ability to add new movies in database as well as deleting them from it and many other options. When user successfully log in, he has ability to delete movies from his watchlist, and can borrow new movies as well as rating them.

**Keywords:** bootstrap, database, movies, video library, website

## ŽIVOTOPIS

Dario Lucić rođen je 4. ožujka 1998. u Slavonskom Brodu. Osnovnu školu pohađao je i završio u Domaljevcu, BiH. Osnovno obrazovanje trajalo je od 2004. do 2012. godine. Godine 2012. upisao je ekonomsku srednju školu koju je pohađao u Orašju, BiH do 2016. godine. Nakon srednje škole upisuje stručni studij informatike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. U trećoj godini odradio je praksu u firmi Atos u Osijeku gdje je usavršavao svoje poznavanje objektno orijentiranog programiranja u programskom jeziku Java. Od stranih jezika odlično poznaje engleski koji mu dodatno pomaže u radu na računalu. Od programskih jezika ima vještine u C-u, Javi i Pythonu, opisnom jeziku HTML-u te jeziku za stiliziranje CSS-u. Upoznat je radom s bazama podataka te ima vještine u različitim web tehnologijama.

---

Potpis: Dario Lucić



## **PRILOZI**

Uz uvezani rad priložen je optički disk s projektnom mapom i dokumentima završnog rada.