**Liceul Teologic Baptist „Emanuel” Oradea**

**The Viewer – Aplicație web pentru prezentarea și gestionarea filmelor**

**Lucrare pentru obținerea atestatului profesional**

**Profil matematică-informatică**

Profesor îndrumător: Absolvent:

Buleandra Constantin Ghiurău David

**2017**

**Cuprins**

**1. Argument 3**

**2. Structura aplicației *The Viewer* 4**

**3. Baze de date 9**

**4. Relații între tabelele unei baze de date 11**

**5. Resurse de hard necesare 12**

**6. Utilizarea aplicației *The Viewer* 13**

**7. Bibliografie 16**

1. **Argument**

**„The Viewer”** este o aplicație care are ca rol principal prezentarea și gestionarea celor mai bune filme din istoria cinematografiei. Paginile Web au fost create în editorul **Sublime Text 3** cu ajutorul limbajelor de programare web (**HTML**, **CSS**, **JavaScript**,  **PHP)** și cu ajutorul librăriilor **(Bootstrap** și **jQuery).** Aplicația a fost testată pe un server local (**Apache**) și cu o bază de date (**MySQL),** acestea au fost create prin intermediul programului **XAMPP.**

1. **Structura aplicației *The Viewer***

În procesul de dezvoltarea a aplicației m-am axat pe ambele părți ale aplicației, **front-end** și **back-end.**

1. **Front-End**

Principala țintă, încă de la primul tag din **HTML**, a fost să fac un design responsive. Cu ajutorul librăriei **Bootstrap** am reușit să creez acest lucru cu foarte multă ușurință. **Bootstrap** este o librărie gratuită și **open-source** care este folosită pentru dezvoltarea și crearea celor mai mari site-uri din lume. **Bootstrap** alături de alte librări pentru front-end precum **Skeletons**, **Foundation** sau **Zebra** au introdus niște standarde pentru design-ul web.

Introducerea librăriilor în paginile **HTML** se face prin tag-ul <**link**></**link**> și <**script**></**script**> în <**head**></**head**> sau în <**body**></**body**> pentru **jQuery** (**JavaScript**). Aplicația **„The Viewer”** are librăriilor descărcate local pentru a facilita timpul de încărcare și pentru a putea fi accesată și fără acces la internet.



Alt instrument folosit pentru crearea unei interfețe mai interactive a fost librăria **jQuery**. Această librărie depăsește limitările impuse de **Vanilla** **JavaScript** și permite programatorului să creeze animații interactive printr-o singură linie de cod. Principalul rol al librăriei **jQuery** a fost de a crea un efect **Parallax**.

**Designul Parallax** constă în crearea unei pagini web în care prin intermediul funcției scroll să se creeze un efect de modificare a elementelor din pagină.

Pe pagina principală, designul **Parallax** se reflectă în 3 elemente. Primul element este titlul care își mărește marginea de sus în funcție de punctul de vizualizare pe pagină. Al doilea element este compus din cele trei rânduri de poze care își schimbă intensitatea luminoasă datorită poziției mouse-lui și punctului de vizualizare. Al treilea element este reprezentat de cele 2 imagini care își schimbă poziția pe pagină prin modificarea coordonatelor de fixare.

Altă funcție creată prin intermediul librăriei **jQuery** este un slideshow de imagini. Funcțiile oferite de slideshow sunt vizibile în imaginea următoare.

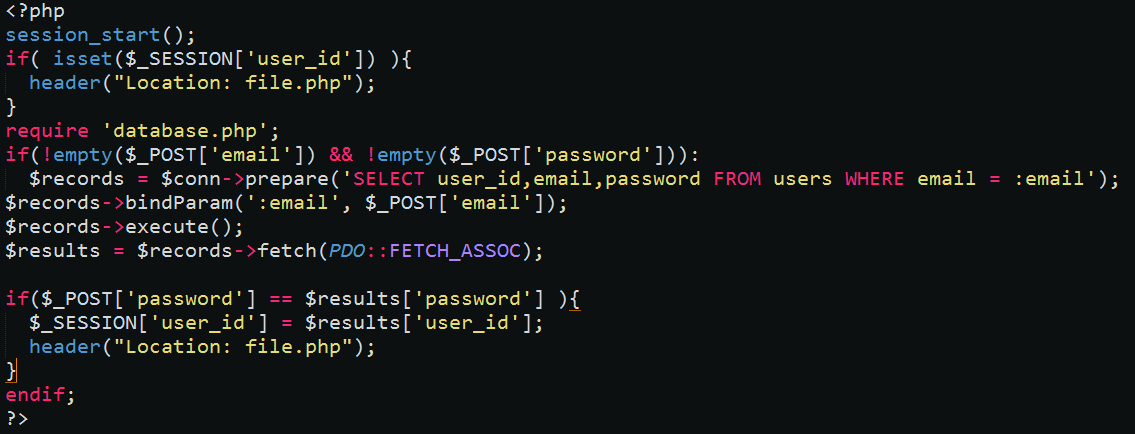
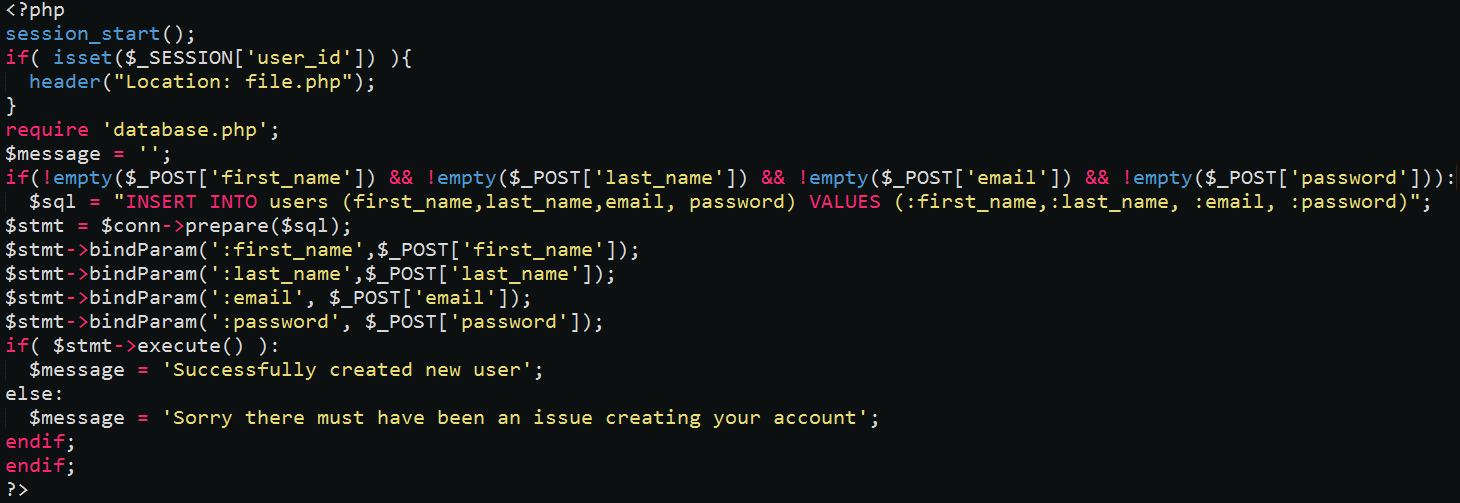
Cea de a treia utilizare a librăriei **jQuery** a constat în crearea unei structuri cu rol de prezentare. Structura cuprinde 12 imagini de format 6vw/8vw, (vw este o metodă de măsurare în css și se calculează 1vw= 1% din lățimea paginii) și un slideshow de dimensiunile 27vw/33vw. Prin intermediul **jQuery** am reușit să creez o tranziție între imagini și slideshow.



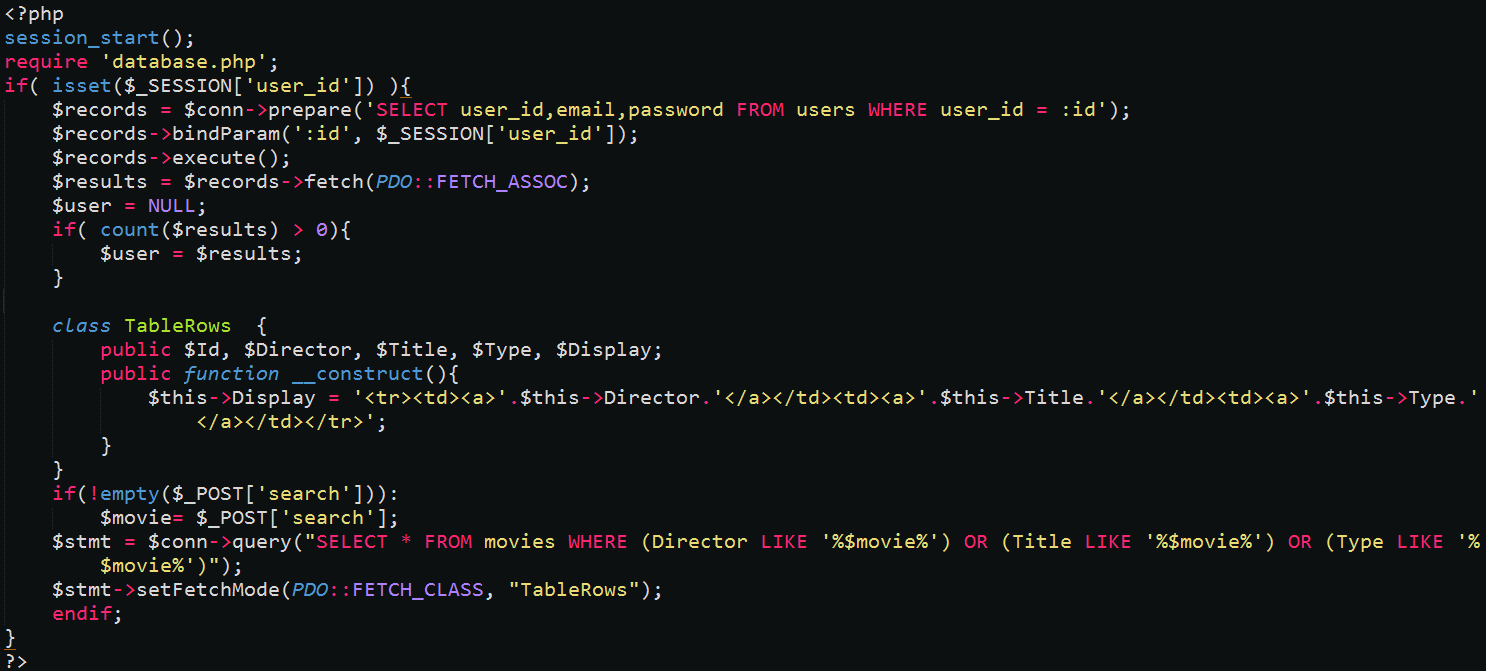
1. **Back-End**

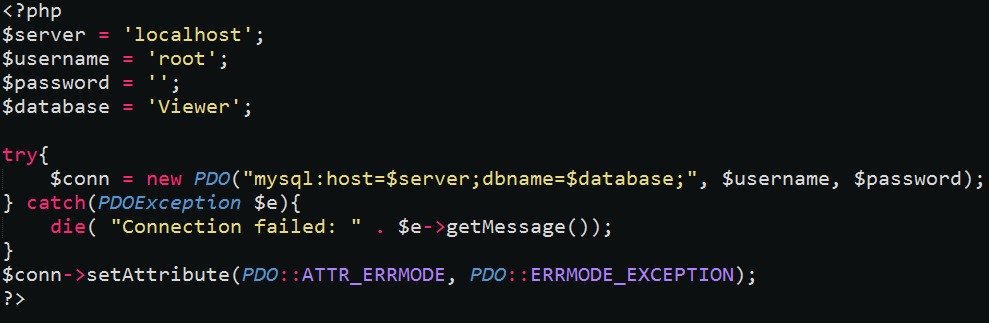
Partea de **Back**-**End** constă majoritar în crearea unor formulare cu ajutorul limbajulul **PHP** și de a transmite datele unei baze de date. Fie pentru prelucrare, căutare, introducere sau ștergere. De asemenea în procesul de creare a formularelor am încercat să elimin principalele vulnerabilități ale formularelor de înregistrare. De aceea se poate observa funcția de îmbinare a parametrilor (**bindParam()).**

Principalele utilizării ale limbajului **PHP**:

1. **Login**
2. **Înregistrare**



1. **Log** **Out**
2. **Afișare**
3. **Crearea conexiunii dintre baza de date și aplicație**



**3.** **Baza de date**

„O bază de date poate fi definită ca ca un ansamblu de date elementare sau structurate, accesibile unei comunități de utilizatori. Mai concret, o bază de date este un ansamblu de fișiere intercorelate, care conține nucleul de date necesare unui sistem informatic.” – Pavel Năstase – *Baze de date – Microsoft Access 2000, Ed. Teora, București, 2000 p.12*

„Bază de date (**BD**) este o colecție de date aflate în relație unele cu altele și structurată astfel încât să poată servi unui anumit scop.

În aplicația **„The Viewer”** baza de date se numește „**viewer”** și este constituită din 3 tabele (**users**, **details**, **movies**).

Creare legăturii dintre baza de date și paginile **HTML** s-a făcut cu ajutorul limbajului **PHP**. Imaginea anterioară este un screenshot a liniilor de cod din fișierul „**database**.**php**”. Cele 4 variabile stochează numele serverului, numele, parola și baza de date. Funcția de conectare la baza de date este cea mai nouă și este recomandată deoarece numărului limitat de vulnerabilități.

Acțiuni care se pot face cu baza de date:

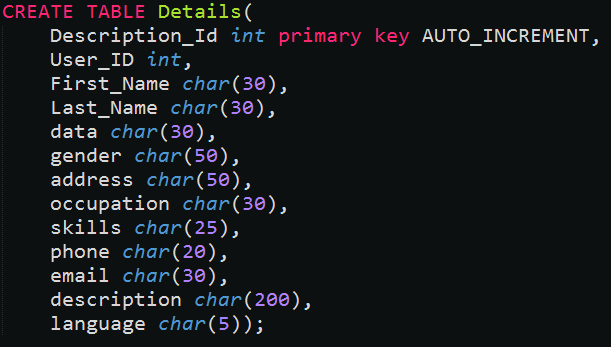
1. **Construirea** Bazei de Date, adică introducerea datelor în baza de date.
2. **Administrarea** Bazei de Date: accesul la date în funcție de drepturile utilizatorilor, asigurarea coerenței bazei de date (recuperarea datelor în urma unor incidente), asigurarea securității datelor stocate.
3. **Interogarea** Bazei de Date, adică extragerea și vizualizarea datelor în funcție de anumite criterii și condiții.
4. **Actualizarea** Bazei de Date, adică modificarea structurii Bazei de Date sau a datelor stocate în Baza de Date.

Codul pentru crearea bazei de date.

1. **Crearea bazei de date**



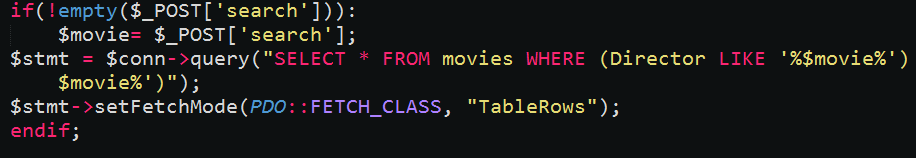
1. **Crearea tabelelor –** fiecare tabelă are o cheie primară și o cheie secundară, astfel încât să faciliteze construirea relațiilor pe baza căruia se vor modifica ambele tabele în momentul introducerii datelor.



1. **Introducerea** **datelor** – mai jos puteți observa un exemplu de date introduse în tabela **„Details”.**



1. **Căutarea în baza de date** – următoarea imagine surprinde codul care acționează atunci când introducem un set de caractere în formularul de căutare a filmelor.

****

**4. Relații între tabelele unei baze de date**

Am menționat că în modelele relaționale datele sunt organizate în tabele, iar între tabele se pot stabili diferite tipuri de relații.

**Relații 1 – 1 (one to one)**

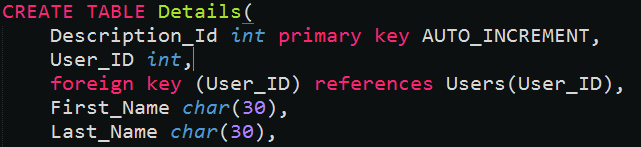
Unei înregistrări din prima tabelă îi corespunde o singură înregistrare din a doua tabelă. Reciproc, unei înregistrări din a doua tabelă îi corespunde o singură înregistrare din prima tabelă. În cazul acesta este necesar ca în cele două tabele să fie folosit același atribut pe post de cheie primară.

**Relații 1 – n (one to many)**

Unei înregistrări din prima tabelă (considerată tabela **master** sau **principală**) îi corespund mai multe înregistrări în a doua tabelă. Unei înregistrări din a doua tabelă (considerată tabela **slave** sau **secundară**) îi corespunde o singură înregistrare în prima tabelă. Relația se stabilește prin intermediul *cheii primare* din tabela principală și al *cheii externe* din tabela secundară.

**Relații n – m (many to many)**

Unei înregistrări din prima tabelă îi corespund mai multe înregistrări în a doua tabelă, iar unei înregistrări din a doua tabelă îi corespund mai multe înregistrări în prima tabelă. Pentru implementarea acestui tip de relație trebuie creată o tabelă suplimentară care să conțină obligatoriu, în calitate de chei externe, atributele care reprezintă cheile primare în tabelele inițiale. Astfel, între prima tabelă și tabela suplimentară se stabilește o relație *one to many*, iar între a doua tabelă și cea suplimentară se stabilește, de asemenea, o relație *one to many*. Putem spune, în sens figurat, că relația inițială *many to many* a fost „spartă” în două relații *one to many*.



**5. Resurse de hard necesare**

Pentru rularea aplicației ***„The Viewer”***  sunt necesare următoarele resurse:

*Resurse minime :*

*Hardware :*

* *Procesor (CPU) :* 500MHz+
* *Memorie (RAM):* 64MB
* *Spaţiu liber pe hard-disk :* 20MB
* *Monitor :* indiferent (design responsive)
* *Un browser oarecare:* Google Chrome ,Mozilla Firefox, Opera

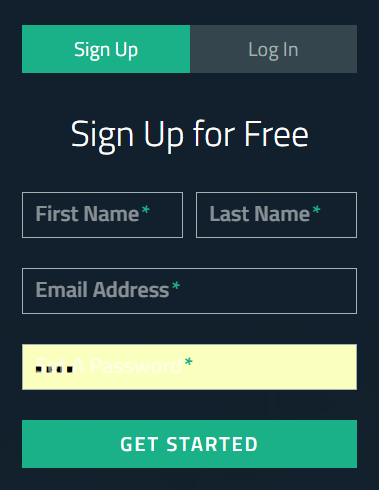
*Resurse recomandate:*

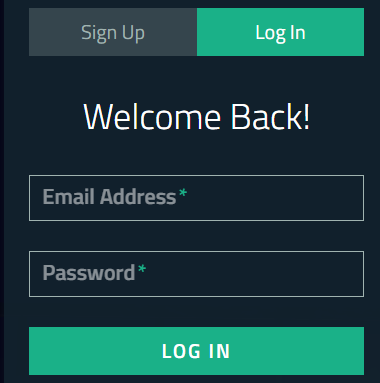
*Hardware :*

* *Procesor (CPU) :* 1000 MHz +
* *Memorie (RAM):* 128 MB
* *Spaţiu liber pe hard-disk :* 20 MB
* *Monitor :* indiferent *(*designresponsive*)*
* *Browser :* GoogleChrome

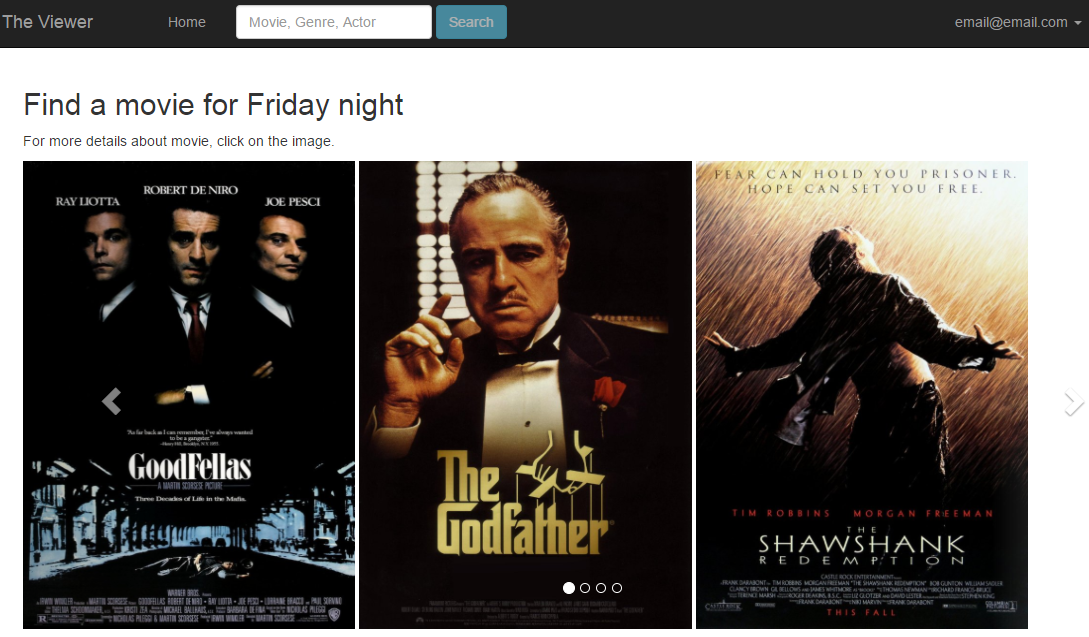
1. **Utilizarea aplicației *The Viewer***

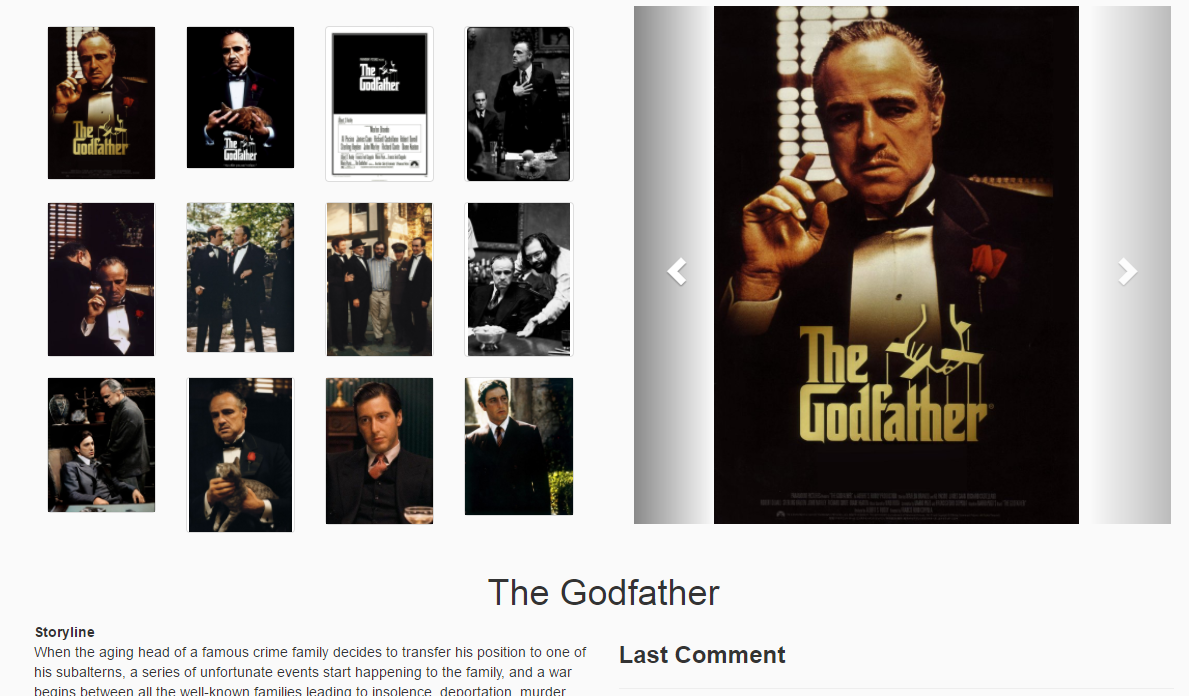
Prima pagină are rol introductiv, aici putem observa designul Parallax și formularul de înscriere. Primul pas este înregistrarea, iar după mesajul de confirmare, vă puteți loga.



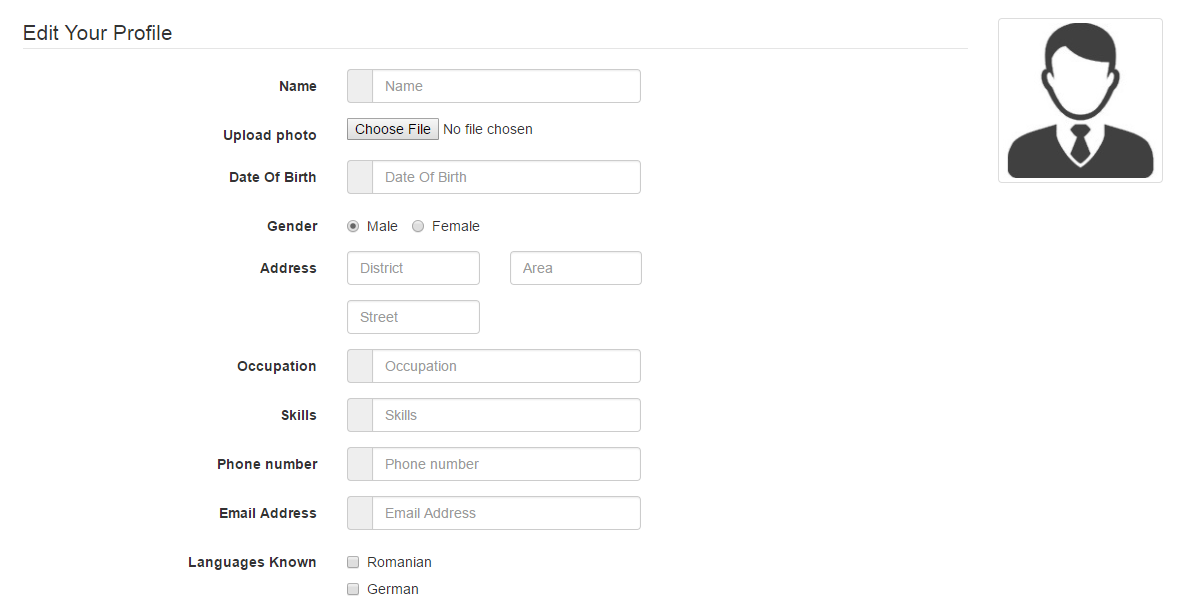


După ce v-ați logat puteți alege un film din slideshow-urile pe categorii sau să introduceți un titlu,director sau gen în formularul de căutare.

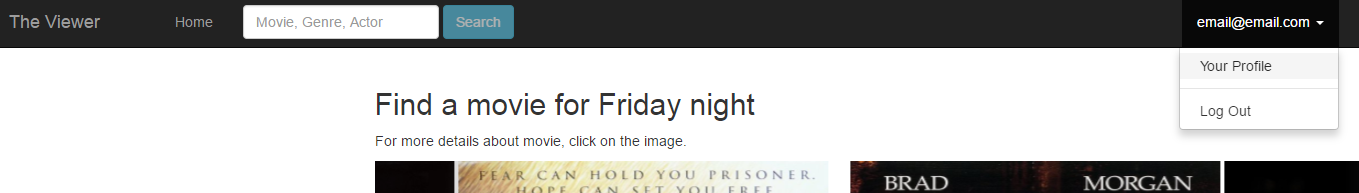
****

Pagina de prezentare a filmului este identică pentru toate filmele.

Contul creat vă oferă și posibilitatea de a adăuga alte detalii despre dumneavoastră.

****

**7.**

După ce ați găsit filmul dorit și v-ați uitat la descrierea acestuia, puteți da log out la pagină pentru a „distruge sesiunea” și de a elibera memoria browserului.

După log out veți fi redirecționat spre pagina principală.

1. **Bibliografie**

Diana Popovici și Ovidiu Popovici*, Proiectare WEB*, Editura Universității din Oradea, 2008

Laura Lemay, Rafe Colburn, Jennifer Kyrnin*, HTML, CSS, JAVASCRIPT WebPublishing*, 2007

Marijn Haverbeke, *Eloquent* *JavaScript*, 2011

Douglas Crockford, *JavaScript*: *The* *Good* *Parts*, 2008

Shay Howe, *Learn* *to* *Code*: *HTML* *&* *CSS*, 2009

<https://developer.mozilla.org>

<https://docs.microsoft.com/>

<https://developers.google.com>

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

<http://stackoverflow.com/>

<http://php.net/manual/ro/>

<https://api.jquery.com>

<http://getbootstrap.com>