Budapesti Műszaki Szakképzési Centrum

Neumann János Informatikai Technikum

***Szakképesítés neve:*** Szoftverfejlesztő

***OKJ száma:*** 54 213 05

**ZÁRÓDOLGOZAT**

**Filmellátó**

Répás Csaba Zsigmond Viktor Árpád

konzulens 14. rsze

Budapest, 2021.

Tartalom

[Bevezető 4](#_Toc65143087)

[Témaválasztás 4](#_Toc65143088)

[A projekt célja 4](#_Toc65143089)

[Felhasználói dokumentáció 4](#_Toc65143090)

[Általános specifikáció, ismertető 4](#_Toc65143091)

[Rendszerkövetelmények 5](#_Toc65143092)

[Első indítás 5](#_Toc65143093)

[Indítás Visual Studio-val 5](#_Toc65143094)

[Indítás Internet Information Services és Microsoft SQL Server Management Studio-val 5](#_Toc65143095)

[Használati útmutató 6](#_Toc65143096)

[Felhasználói felület 6](#_Toc65143097)

[Felhasználói jogosultságok 8](#_Toc65143098)

[Dolgozói felhasználói fiók 9](#_Toc65143099)

[Tulajdonos (admin) felhasználói fiók 10](#_Toc65143100)

[Ügyfél hozzáadása 11](#_Toc65143101)

[Ügyfél adatainak megtekintése 11](#_Toc65143102)

[Ügyfél adatainak módosítása 12](#_Toc65143103)

[Ügyfél blokkolása 12](#_Toc65143104)

[Ügyfél törlése 13](#_Toc65143105)

[Film hozzáadása 13](#_Toc65143106)

[Film adatainak megtekintése 14](#_Toc65143107)

[Film adatainak módosítása 14](#_Toc65143108)

[Film törlése 15](#_Toc65143109)

[Dolgozó hozzáadása 15](#_Toc65143110)

[Dolgozók listájának megtekintése 16](#_Toc65143111)

[Dolgozó törlése 16](#_Toc65143112)

[Film bérbeadása 17](#_Toc65143113)

[Kibérelt filmek megtekintése 17](#_Toc65143114)

[Kibérelt film visszavétele 18](#_Toc65143115)

[Fejlesztői dokumentáció 19](#_Toc65143116)

[Témamegjelölés 19](#_Toc65143117)

[Rendszerterv 19](#_Toc65143118)

[Jogosultsági szintek 20](#_Toc65143119)

[Adatbázis terv 21](#_Toc65143120)

[Az ügyfelekről eltárolandó információk 22](#_Toc65143121)

[A dolgozókról eltárolandó információk 22](#_Toc65143122)

[A filmekről eltárolandó információk 22](#_Toc65143123)

[A kiadott filmekről (bérlések) eltárolandó információk 22](#_Toc65143124)

[Customers (ügyfelek) tábla: 23](#_Toc65143125)

[MembershipTypes (ügyféltagságok fajtái) tábla: 24](#_Toc65143126)

[Movies (filmek) tábla: 24](#_Toc65143127)

[Genres (filmtípusok) tábla: 25](#_Toc65143128)

[Rentals (filmbérlések) tábla: 25](#_Toc65143129)

[Felhasznált eszközök, technológiák 26](#_Toc65143130)

[C# programozási nyelv 26](#_Toc65143131)

[ASP.NET MVC keretrendszer 27](#_Toc65143132)

[Entity Framework 29](#_Toc65143133)

[Internet Information Services (IIS) 29](#_Toc65143134)

[jQuery 29](#_Toc65143135)

[Bootstrap 29](#_Toc65143136)

[Tesztelési dokumentáció 29](#_Toc65143137)

[Webalkalmazás adatbázisműveletek tesztelése 29](#_Toc65143138)

[Felhasználói felület (GUI) tesztelése 29](#_Toc65143139)

[Összegzés 29](#_Toc65143140)

[Amit sikerült megvalósítani 29](#_Toc65143141)

[Fejlesztési lehetőségek 29](#_Toc65143142)

[Személyes tapasztalatok 29](#_Toc65143143)

[Felhasznált források 29](#_Toc65143144)

[Köszönetnyilvánítás 29](#_Toc65143145)

[Ábrajegyzék 29](#_Toc65143146)

# Bevezető

## Témaválasztás

Záródolgozatom témájául egy „régimódi” videotéka menedzselésére szolgáló weboldal elkészítését választottam, mivel jómagam film- illetve sorozatrajongó vagyok. A projekt egy összesen 11 táblából álló adatbázist használ, melybe beleértendőek az ASP.NET Framework által automatikusan létrehozott táblák is.

Záródolgozatom témájának kiválasztásánál elsősorban a kíváncsiság vezérelt. Mindig is érdekelt, hogy miből áll össze, valamint hogyan működik egy komplexebb weboldal. Középiskolai tanulmányaimból kifolyólag a programozást mindig is C# nyelven tanultam, valamint az OKJ képzés keretein belül is ezt a nyelvet használtuk leginkább, így kézenfekvő volt, hogy záródolgozatom is ebben készítsem el. Ezeket összerakva jutottam arra a döntésre, hogy elképzeléseim megvalósítására a C# alapú, MVC technológiát használó ASP.NET Framework lesz a legjobb választás számomra.

## A projekt célja

Célom egy olyan rendszer megvalósítása volt, ahol a videotéka tudja menedzselni ügyfeleit, dolgozóit, éppen kibérelhető filmjeit, valamint magukat a bérléseket. Fontos, hogy maga a weboldal az interneten keresztül bárki számára elérhető, hogy látható lehessen a kínálat, az elérhetőségi adatok, és egy rövid bemutatkozás, viszont felhasználói fiókba bejelentkezni csakis kizárólag a dolgozók, és a tulajdonos tudnak.

A dolgozók tudnak új ügyfelet beregisztrálni, már meglévő ügyfél adatait megtekinteni, módosítani, és ügyfelet törölni. Megtekinthetik a bérlések állapotát, új bérléseket hozhatnak létre, továbbá filmeket vissza tudnak venni, azaz a bérléseket módosíthatják. Ezen felül természetesen látják a filmek teljes listáját.

A tulajdonos szintén tudja az előzőek mindegyikét. Továbbá ő tud új dolgozót regisztrálni, törölni, valamint megtekinteni a dolgozók listáját. A tulajdonos tud új filmet felvenni a kínálatba, törölni, illetve módosítani már meglévő filmek adatait.

# Felhasználói dokumentáció

## Általános specifikáció, ismertető

A Filmellátó weboldal egy magyarországi videotéka mindennapi feladatainak menedzselését lehetővé tévő rendszer. Ennek segítségével követhetőek a filmbérlések információi, a raktárkészleti információk, valamint regisztrálhatóak dolgozók, ügyfelek és persze új filmek is.

Fontos, hogy a felhasználói fiókok felhasználóneveit, valamint jelszavait soha ne adjuk ki külsős személyeknek, ugyanis az visszaélésre adhat okot. Bejelentkezés nélkül természetesen látható a filmek listája, információk a cégről, valamint az elérhetőségi adatok, mint például e-mail címek, telefonszám, közösségi média platformok, Google Maps meghatározás, de minden más funkció a bejelentkezés után válik elérhetővé.

## Rendszerkövetelmények

A weboldal használatához mindenképp szükség lesz internetkapcsolatra, valamint valamilyen webböngészőre, például Google Chrome / Mozilla Firefox / Microsoft Edge / Opera / Vivaldi.

A weboldal működtetéséhez Visual Studio 2010 (vagy újabb), vagy Internet Information Services (IIS) és Microsoft SQL Server Management Studio használata szükséges.

## Első indítás

### Indítás Visual Studio-val

A projekt mappájában található egyetlen Filmellato.sln nevű fájlt kell megnyitni a Visual Studio segítségével. Ezután a fenti menüsoron található zöld nyíllal tudjuk elindítani a Filmellátó weboldalt, mely az alapértelmezett böngészőben fog megnyílni.

### Indítás Internet Information Services és Microsoft SQL Server Management Studio-val

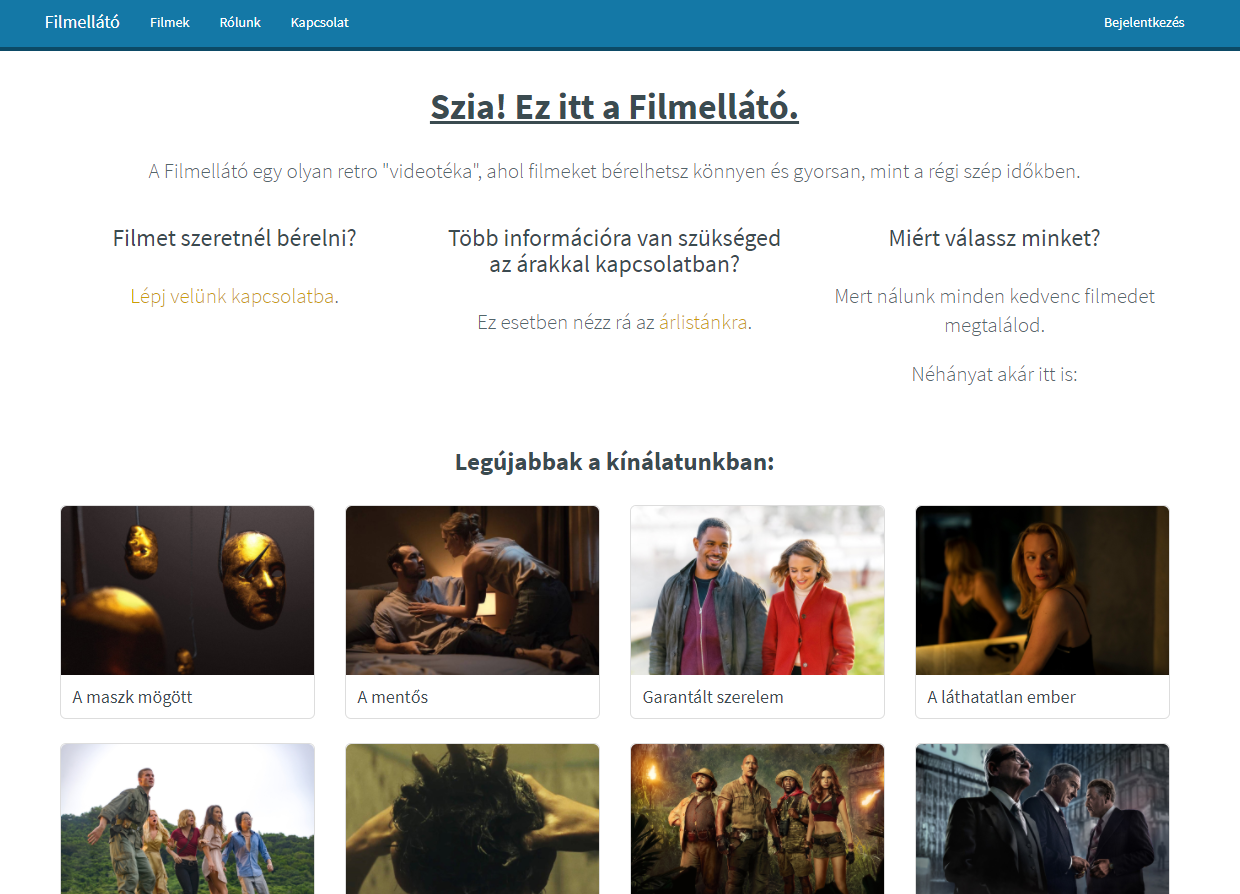
A mellékelt filmellato\_dokumentumok mappában található üres, vagy feltöltött adatbázis .mdf és .ldf kiterjesztésű fájljait kell beimportálni a Microsoft SQL Server Management Studio programba.

Ezután a projekt Web.config állományában kell módosítani a <connectionString></connectionString> kulcs-érték párt úgy, hogy az előzőleg importált adatbázis connectionString-jét tartalmazza.

Majd hozzá kell adnunk egy új weboldalt az Internet Information Services segítségével, ahol meg kell adnunk egy host nevet, majd ki kell választanunk a Filmellato-master mappa elérési útvonalát. Ezeket végrehajtva a weboldal elérhetővé válik a megadott host címén.

## Használati útmutató

### Felhasználói felület

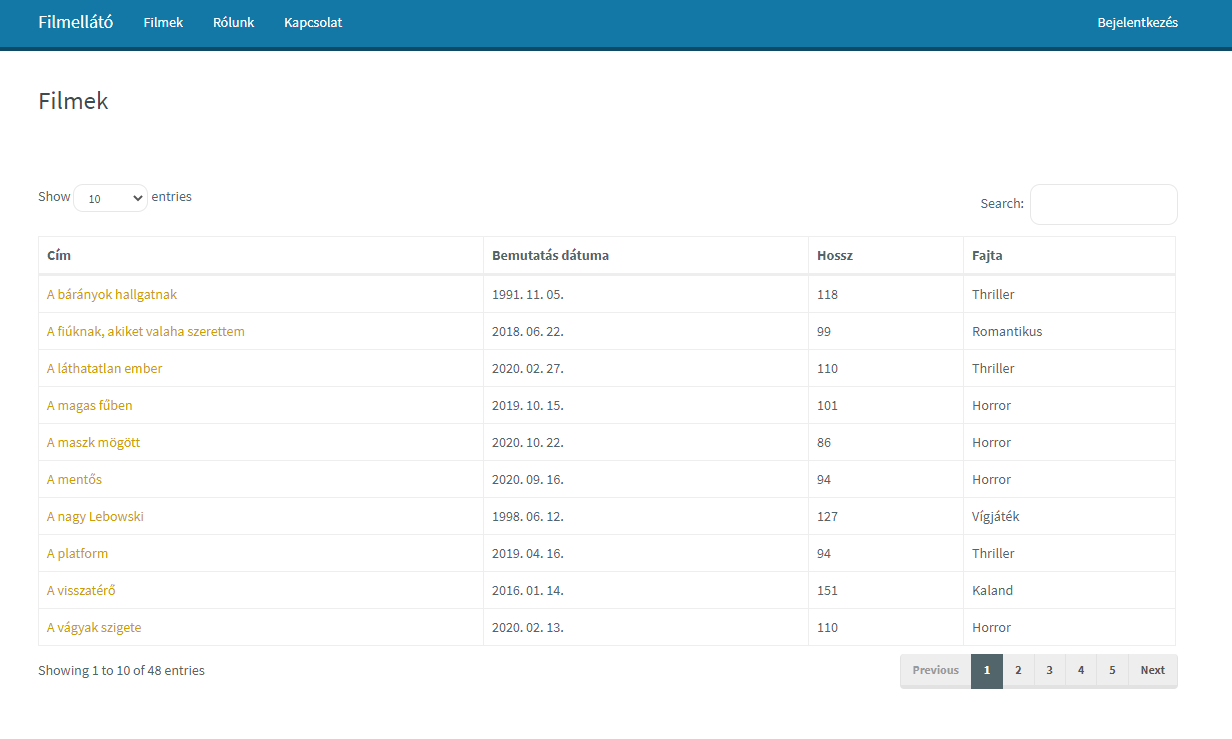
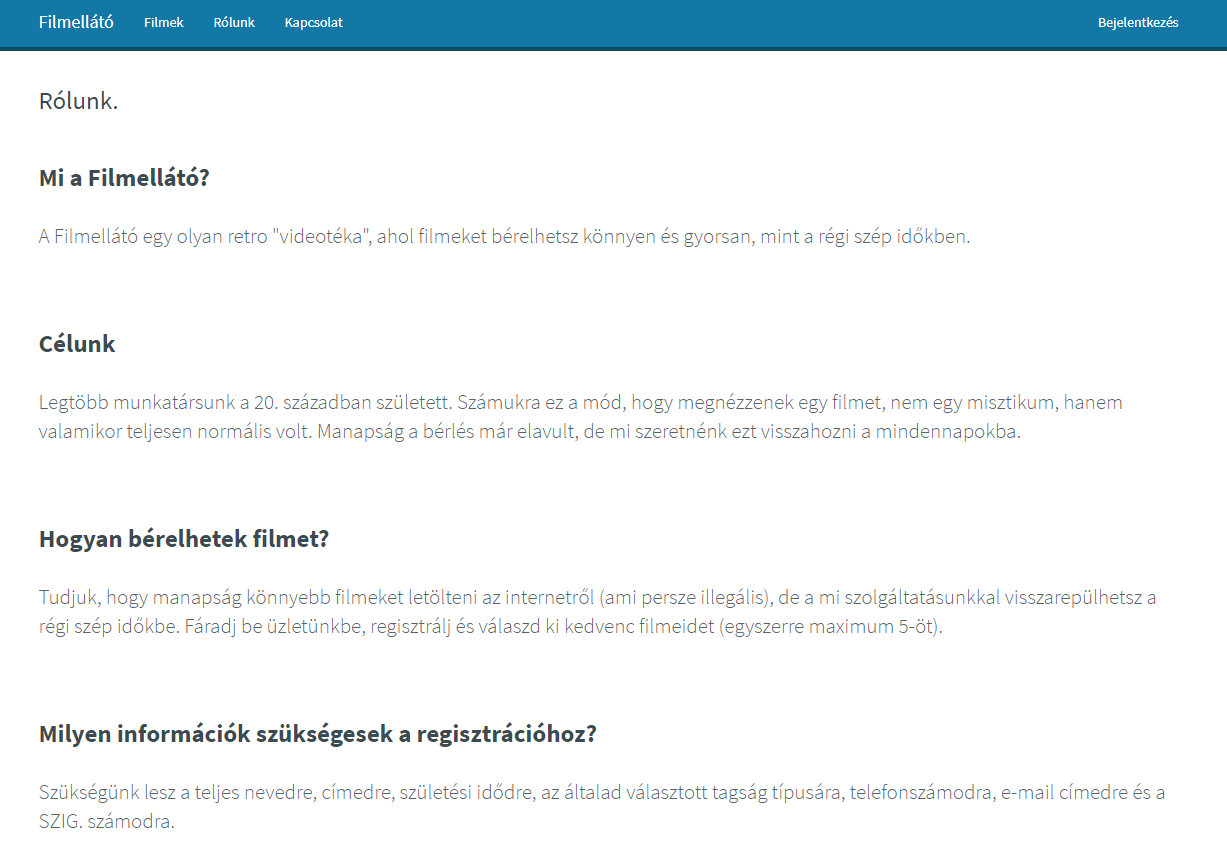
A weboldal indítása után a főoldalra érkezve az alábbi látvánnyal szembesülünk:

2. ábra: Főoldal feltöltött filmekkel

1. ábra: Főoldal feltöltött filmek nélkül

Az 1. ábrán láthatjuk a főoldal kinézetét a 0. indításnál, amikor még nincsen feltöltve film az adatbázisba. Amikor már elegendő film került feltöltésre (legalább 8), a 2. ábrán látható módon fog kinézni a főoldal. A többi kategória (Családi, Romantikus, Horror, Vígjáték) megjelenéséhez a főoldalon szintén mindegyikből legalább 8 darabot kell feltöltenünk, ezután pedig mindig a legfrissebb 8 lesz látható.

Az oldal tetején találjuk a menüt. Amennyiben a „Filmellátó” feliratra kattintunk, a fentebb látható főoldalra fogunk visszajutni. Ha a „Filmek”, „Rólunk” vagy a „Kapcsolat” menüpontra kattintunk, a 3., 4. és 5. ábrán található látvánnyal találjuk szembe magunkat:



4. ábra: „Rólunk" menüpont

3. ábra: „Filmek” menüpont

### Felhasználói jogosultságok

5. ábra: „Kapcsolat" menüpont

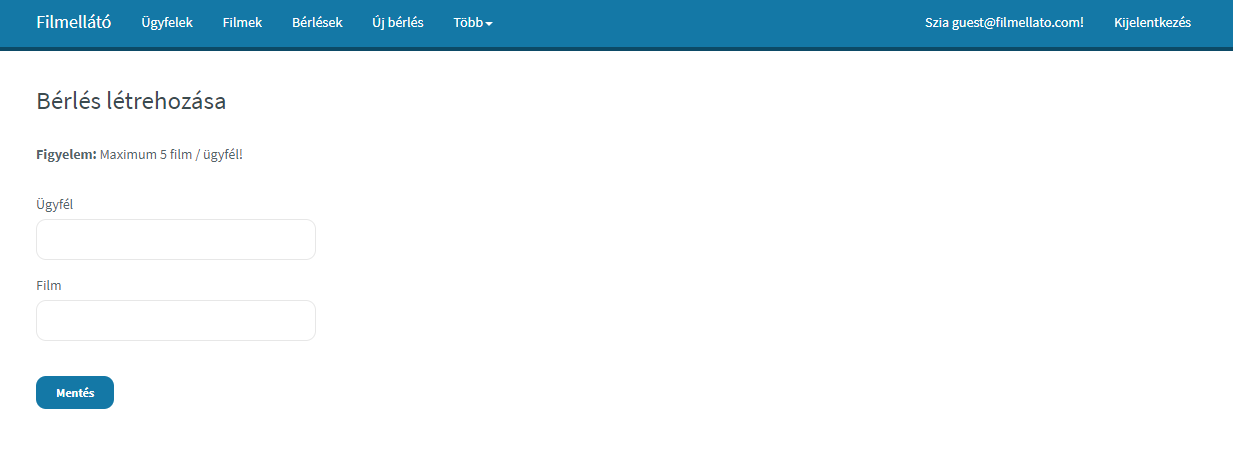
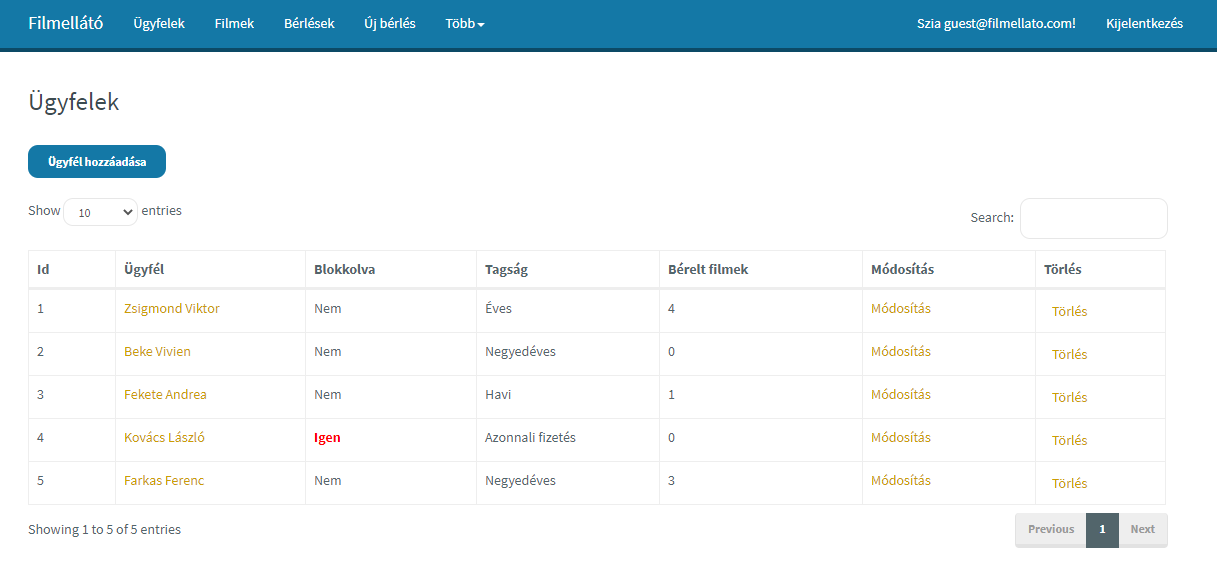
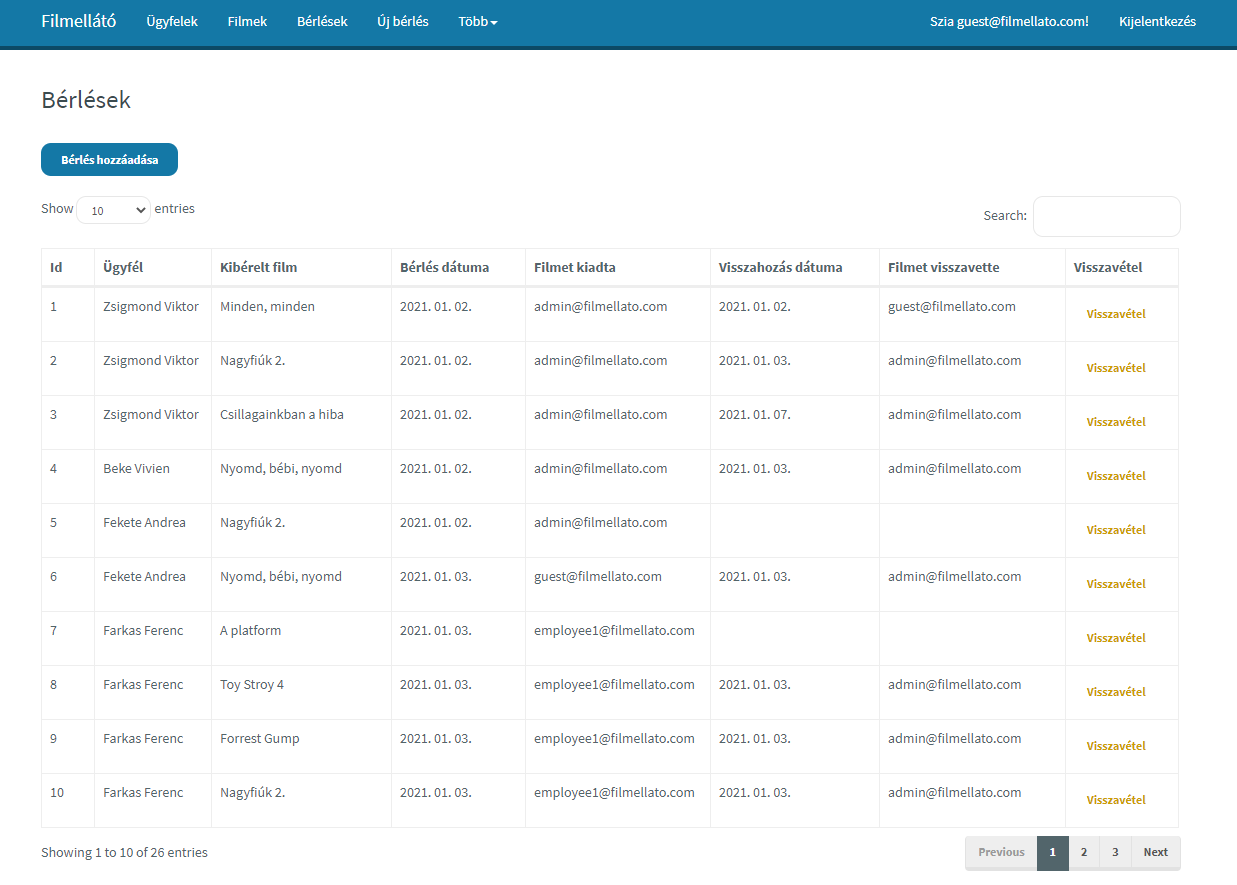
Alapvetően két különböző felhasználói típus van: dolgozó és tulajdonos (admin). Az első indítás során 2 felhasználói fiók kerül automatikusan létrehozásra:

6. ábra: Automatikusan létrehozott felhasználók

|  |  |
| --- | --- |
| **Felhasználónév (e-mail)** | **Alap jelszó** |
| [admin@filmellato.com](mailto:admin@filmellato.com) | Test12345678. |
| [guest@filmellato.com](mailto:guest@filmellato.com) | Test12345678. |

A felhasználói fiókok jelszavát természetesen lehet változtatni, ezt a menüsor jobb felső sarkában az éppen bejelentkezett fiók nevére kattintva érjük el.

### Dolgozói felhasználói fiók

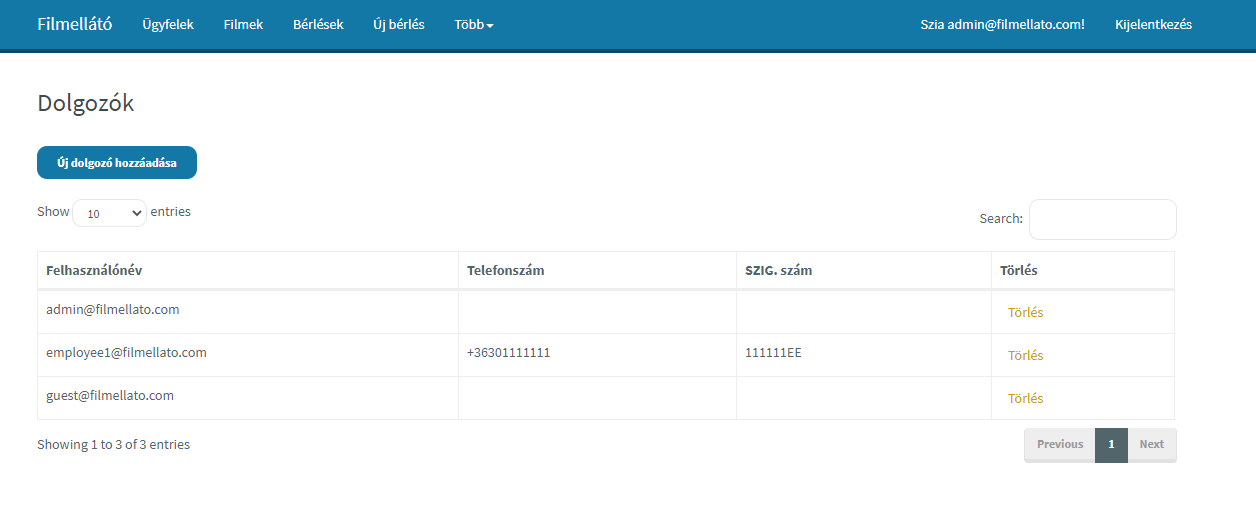
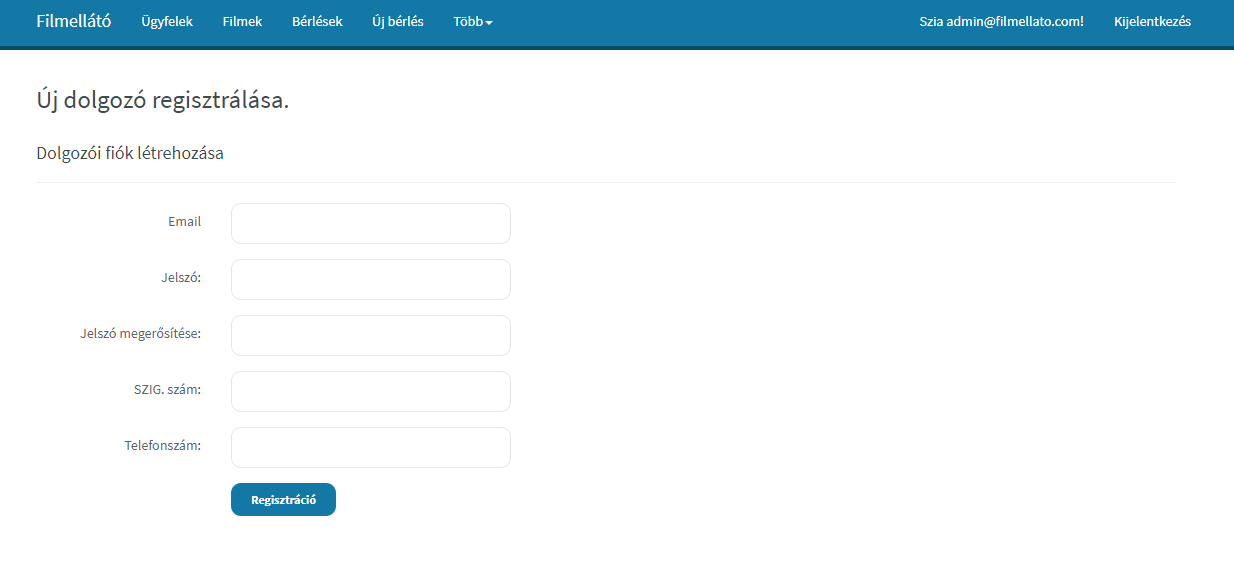
Az alapértelmezetten létrehozott dolgozói fiókba belépve (melynek belépési adatai a 6. ábrán találhatóak) 3 új menüpontot találunk: „Ügyfelek”, „Bérlések”, „Új bérlés”:

8. ábra: „Bérlések” menüpont

9. ábra: „Új bérlés” menüpont

7. ábra: „Ügyfelek” menüpont

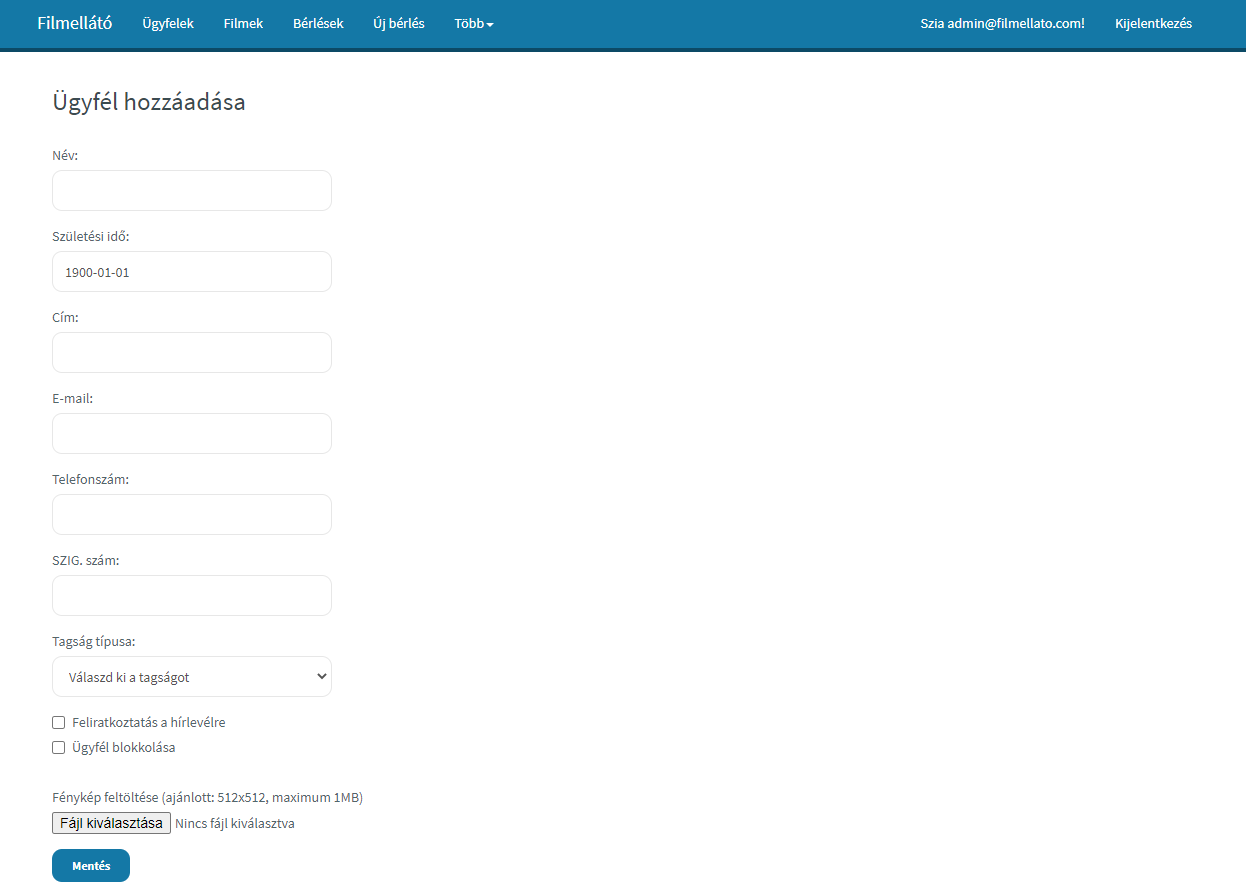
### Tulajdonos (admin) felhasználói fiók

Az alapértelmezetten létrehozott tulajdonos (admin) fiókba belépve (melynek belépési adatai a 6. ábrán találhatóak) 2 új menüpontot találunk a legördülő menüben: „Dolgozók listája”, „Új dolgozó hozzáadása”:

11. ábra: „Új dolgozó hozzáadása” menüpont

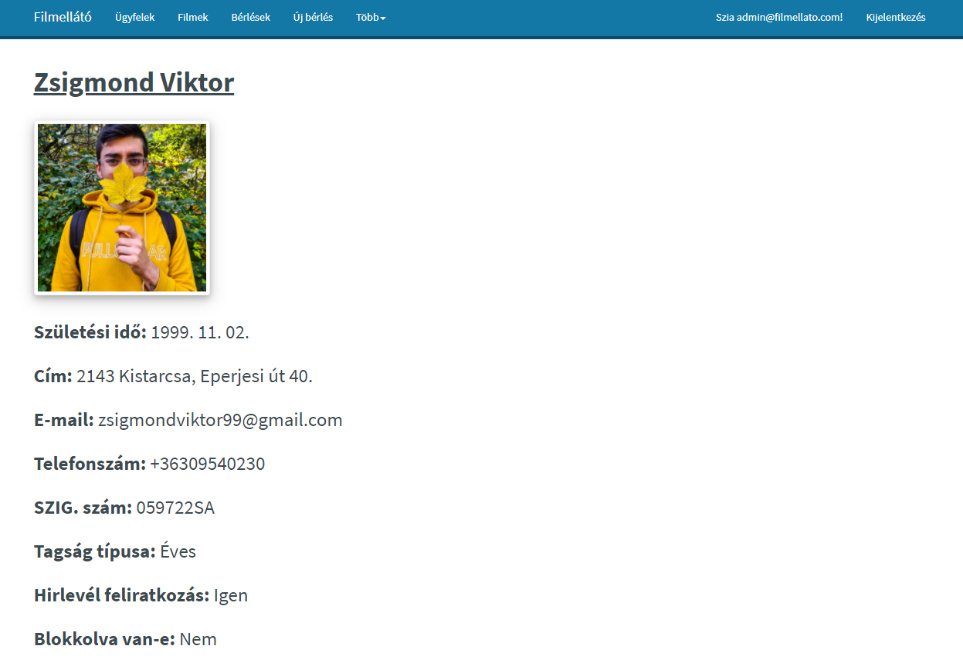
10. ábra: „Dolgozók listája” menüpont

### Ügyfél hozzáadása

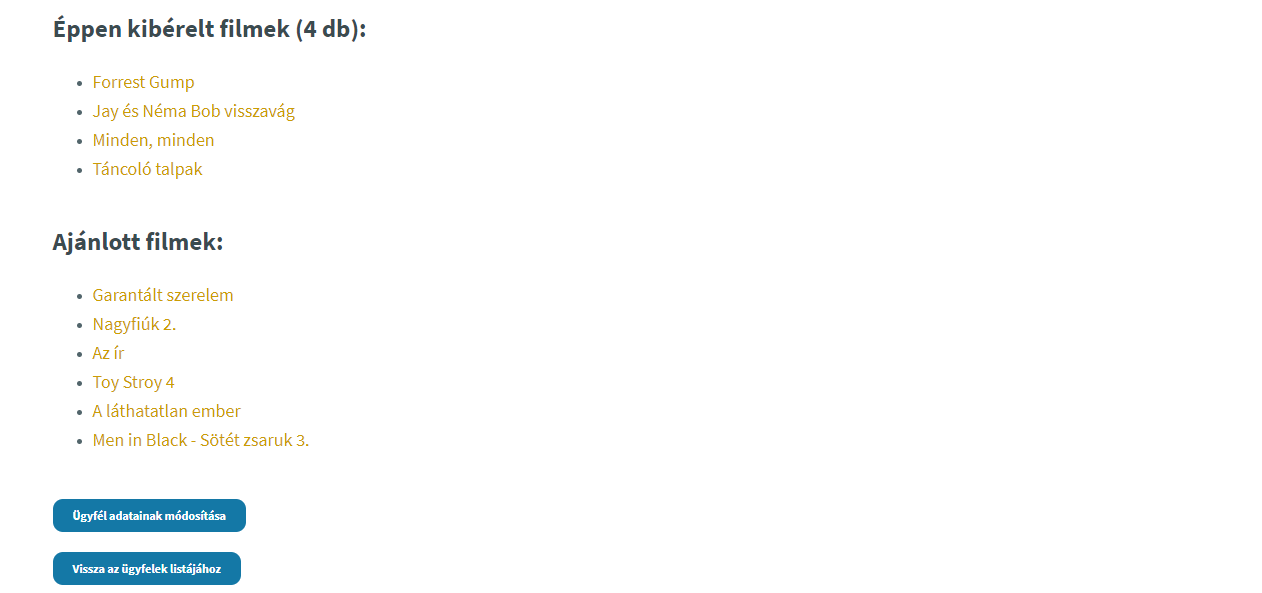
Ügyfél felvétele a dolgozói és a tulajdonosi felhasználói fiókból is lehetséges az „Ügyfelek” menüpont alatt található „Ügyfél hozzáadása” gomb megnyomásával, amennyiben az ügyfél már betöltötte a 18. életévét (kivétel ez alól az „Azonnali fizetés” tagság):

12. ábra: „Új ügyfél hozzáadása” menüpont

### Ügyfél adatainak megtekintése

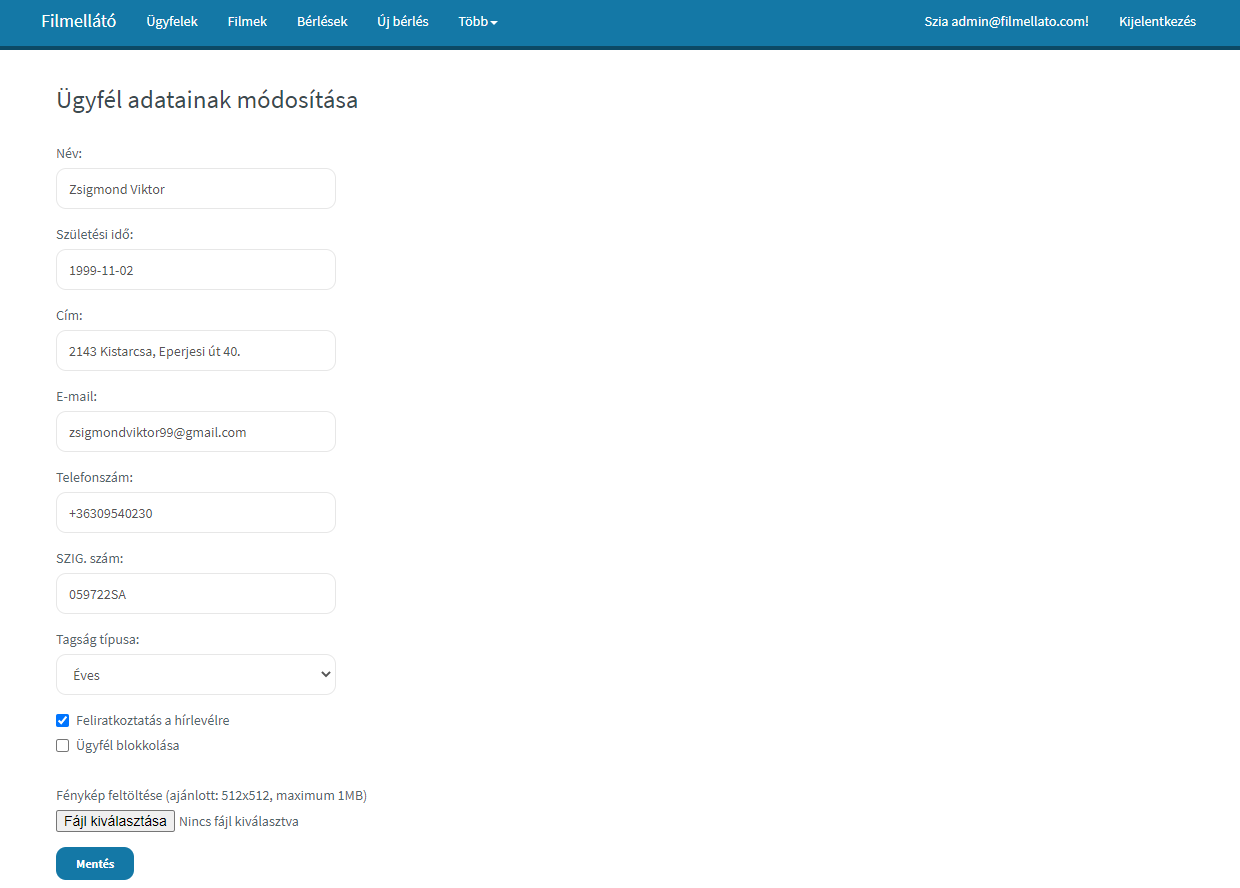
Az ügyfelek adatainak megtekintésére a dolgozói és a tulajdonosi fiókból is van lehetőség. Az „Ügyfelek” menüpont alatt amennyiben rákattintunk egy ügyfél nevére, megtekinthetjük annak minden regisztrált adatát:

13. ábra: Ügyfél adatainak megtekintése

Az adatok alatt megtekinthetjük, hogy az ügyfélnél éppen mely filmek vannak kibérelve, továbbá egy filmajánlót is találunk, mely az ügyfél korábbi bérlései alapján kiválogatott filmeket tartalmaz:

14. ábra: Ügyfél által bérelt filmek, filmajánló

### Ügyfél adatainak módosítása

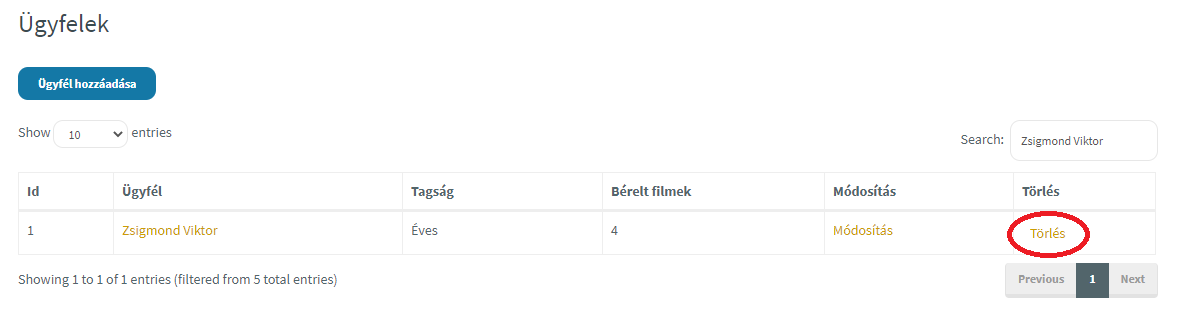
Az ügyfelek adatainak módosítására a dolgozói és a tulajdonosi (admin) fiókból is van lehetőség. Az „Ügyfelek” menüpont alatt amennyiben rákattintunk egy ügyfél neve mellett található „Módosítás” gombra, vagy az ügyfél adatainak oldalán alul, az alábbi látvánnyal szembesülünk:

15. ábra: Ügyfél adatainak módosítása

### Ügyfél blokkolása

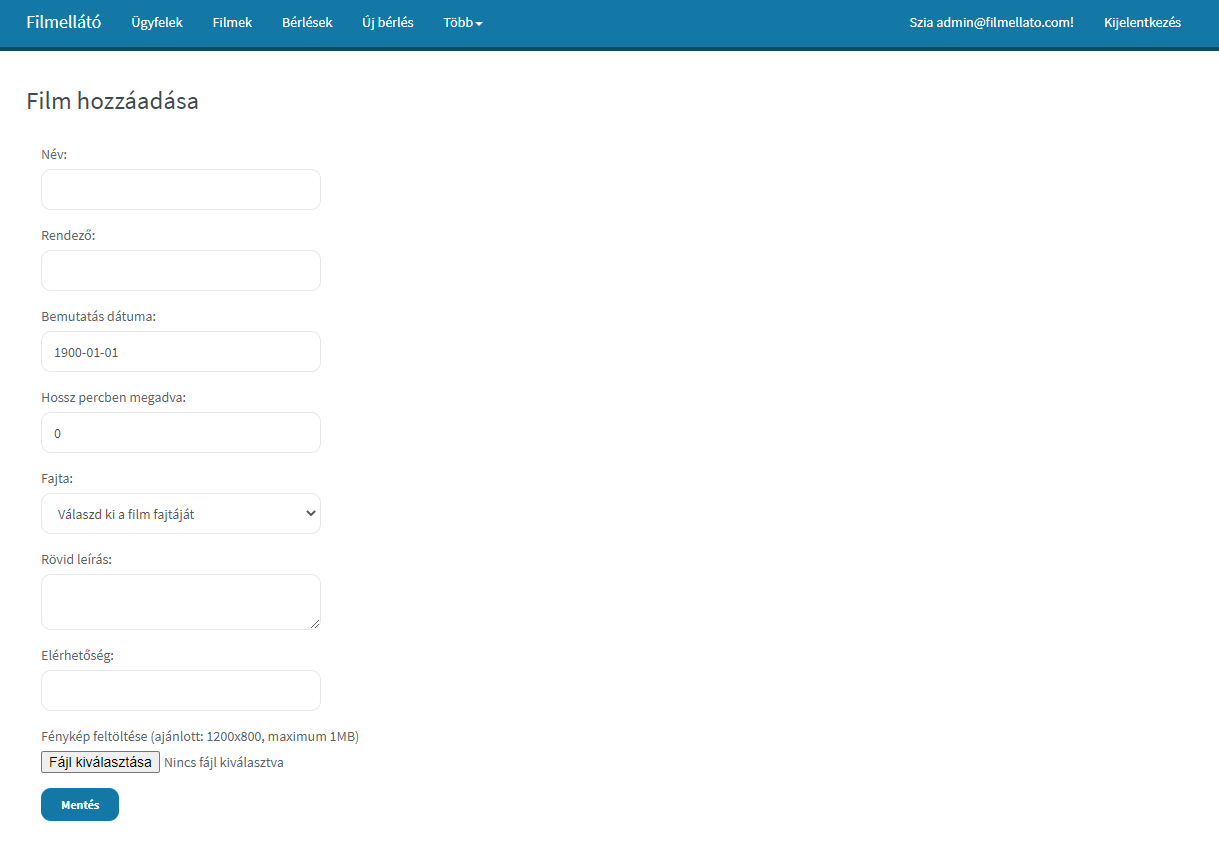
Amennyiben egy ügyfél adatainak módosításánál kiválasztjuk az „Ügyfél blokkolása” lehetőséget, az ügyfél neve mellett megjelenik a „(Blokkolva)” felirat, az ügyfél nem fog tudni filmet bérelni, valamint az adatainak megtekintésénél nem lesz filmajánló a számára. A blokkolás hatására természetesen a jelenlegi bérlései nem tűnnek el, azokat visszahozhatja. A blokkolás később feloldható.

### Ügyfél törlése

Ügyfél törlésére a dolgozói és a tulajdonosi fiókból is van lehetőség, amennyiben az ügyfélnél nincsen 1db kibérelt film sem. Ezt az „Ügyfelek” menüpont alatt az ügyfél neve melletti „Törlés” gombbal tehetjük meg.

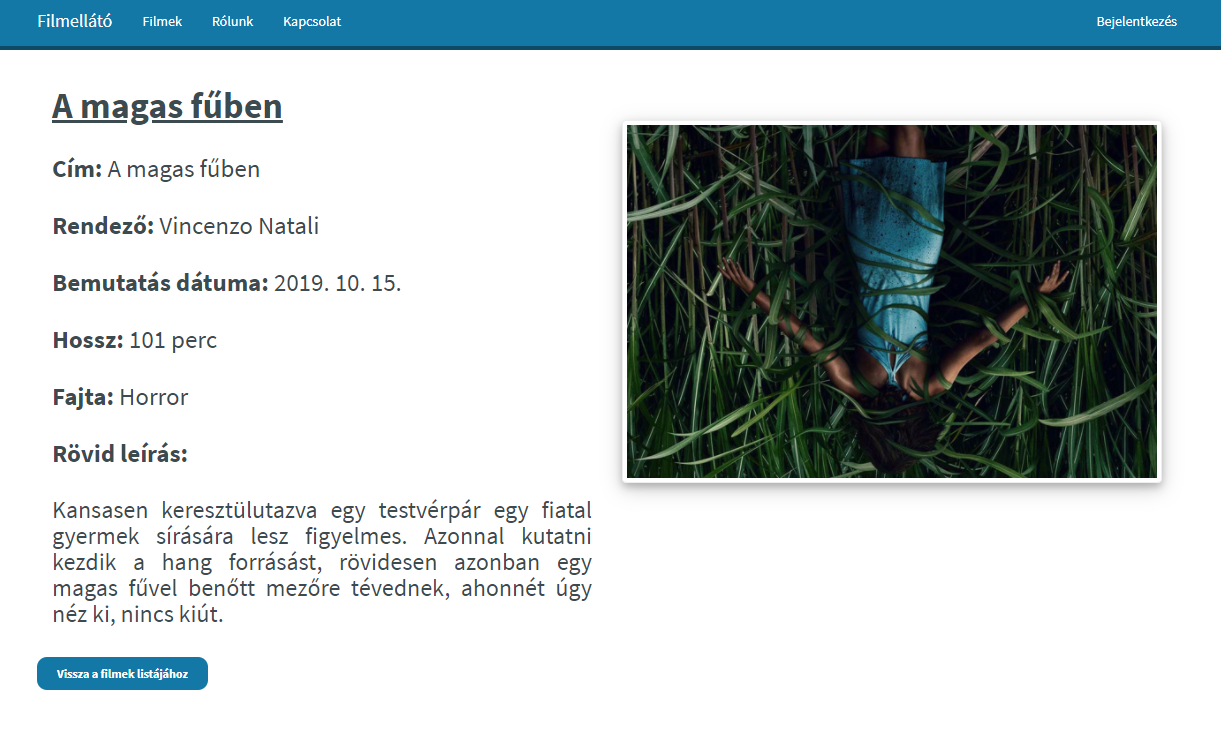
16. ábra: Ügyfél törlése

### Film hozzáadása

Film felvétele csak a tulajdonosi (admin) felhasználói fiókból lehetséges a „Filmek” menüpont alatt található „Film hozzáadása” gomb megnyomásával:

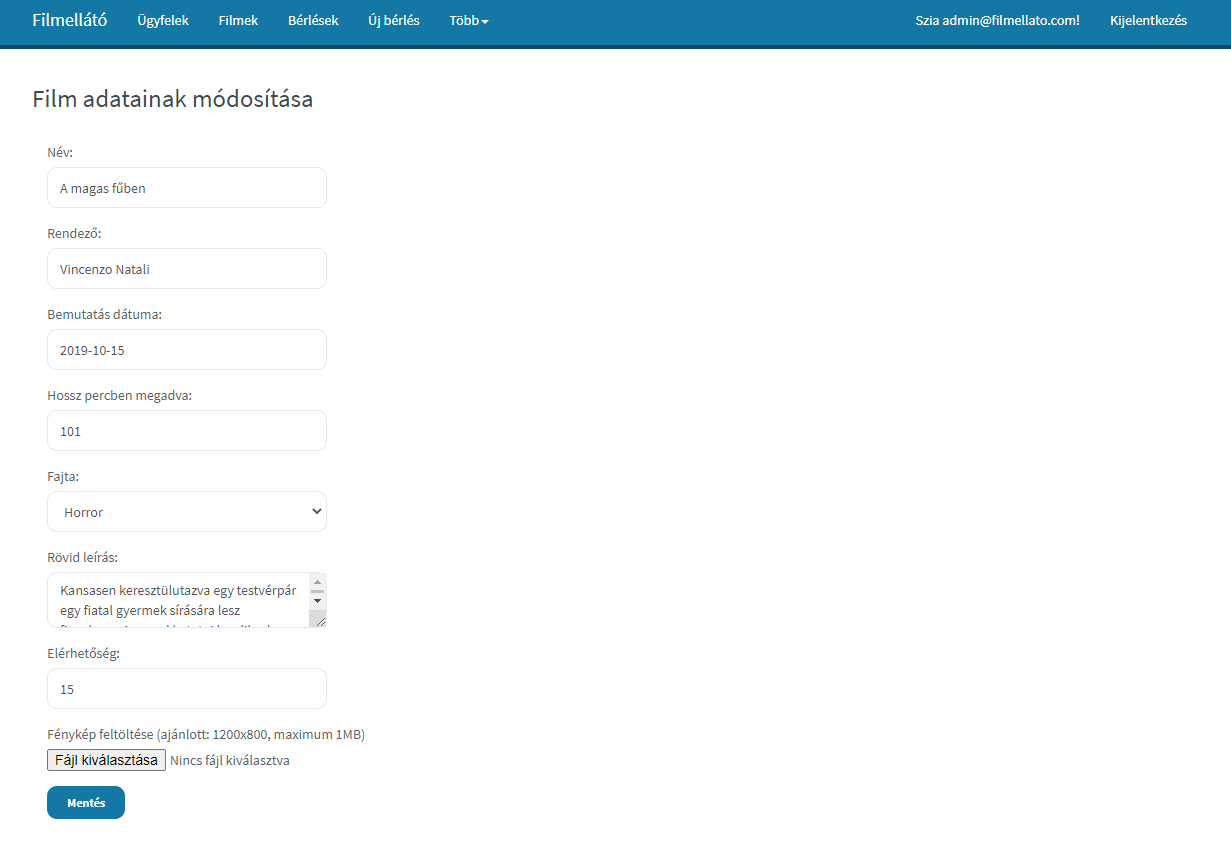
17. ábra: „Film hozzáadása” menüpont

### Film adatainak megtekintése

A filmek listáját, valamint a filmek adatait természetesen a dolgozói, valamint a tulajdonosi (admin) fiókból is meg lehet tekinteni, továbbá a nem regisztrált látogatók is látják ezeket. Amennyiben a „Filmek” menüpont alatt az egyik film címére kattintunk, megtekinthetjük annak minden adatát:

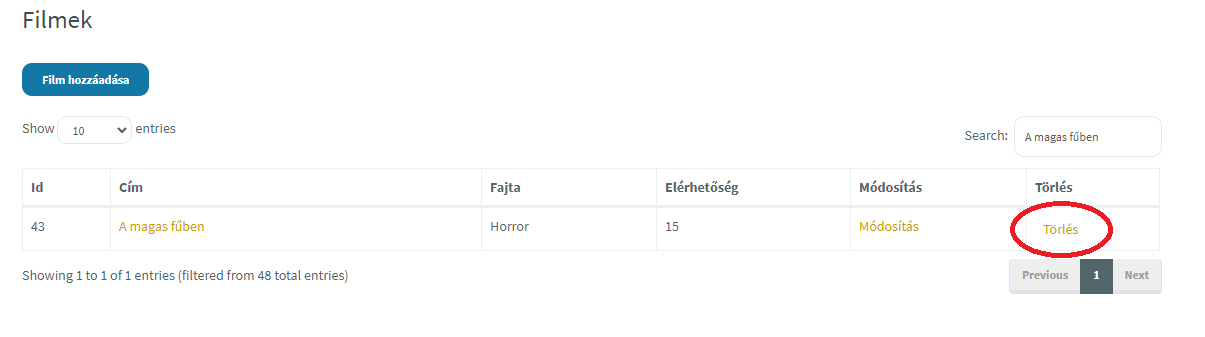
18. ábra: Film adatainak megtekintése

### Film adatainak módosítása

Film adatainak módosítására csak a tulajdonosi (admin) felhasználói fiókból van lehetőség. Amennyiben a „Filmek” menüpont alatt rákattintunk a film címe mellett található „Módosítás” gombra, vagy a film adatainak oldalán alul (ezek csak tulajdonosként belépve jelennek meg), az alábbi látvánnyal szembesülünk:

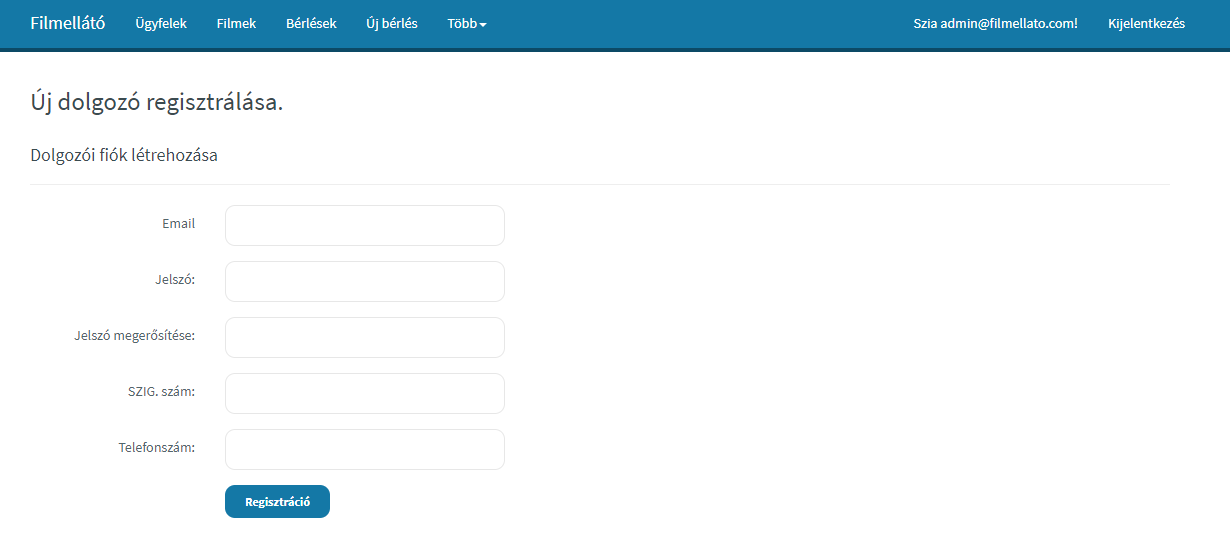
19. ábra: Film adatainak módosítása

### Film törlése

Film törlésére csak a tulajdonosi (admin) felhasználói fiókból van lehetőség. A „Filmek” menüpont alatt a film címe mellett található „Törlés” gombra kattintva érjük el ezt a funkciót.

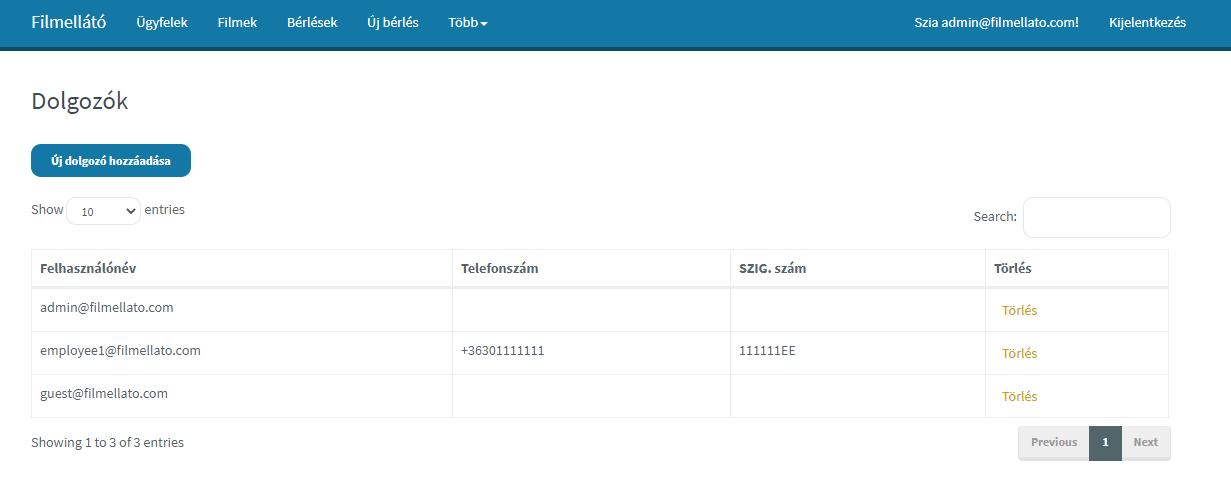
20. ábra: Film törlése

### Dolgozó hozzáadása

Új dolgozó felvételére csak a tulajdonosi (admin) felhasználói fiókból van lehetőség, melyet a „Több” legördülő menüben érünk el az „Új dolgozó hozzáadása” menüpont alatt. Új dolgozó esetén kötelezően ki kell töltenünk minden adatot (e-mail cím (mely a felhasználónév is), jelszó, SZIG (személyi igazolvány) szám, továbbá telefonszám).

21. ábra: „Új dolgozó hozzáadása” menüpont

### Dolgozók listájának megtekintése

A dolgozók listáját csak a tulajdonosi (admin) felhasználói fiókból van lehetőség megtekinteni, melyet a „Több” legördülő menüben érünk el a „Dolgozók listája” menüpont alatt.

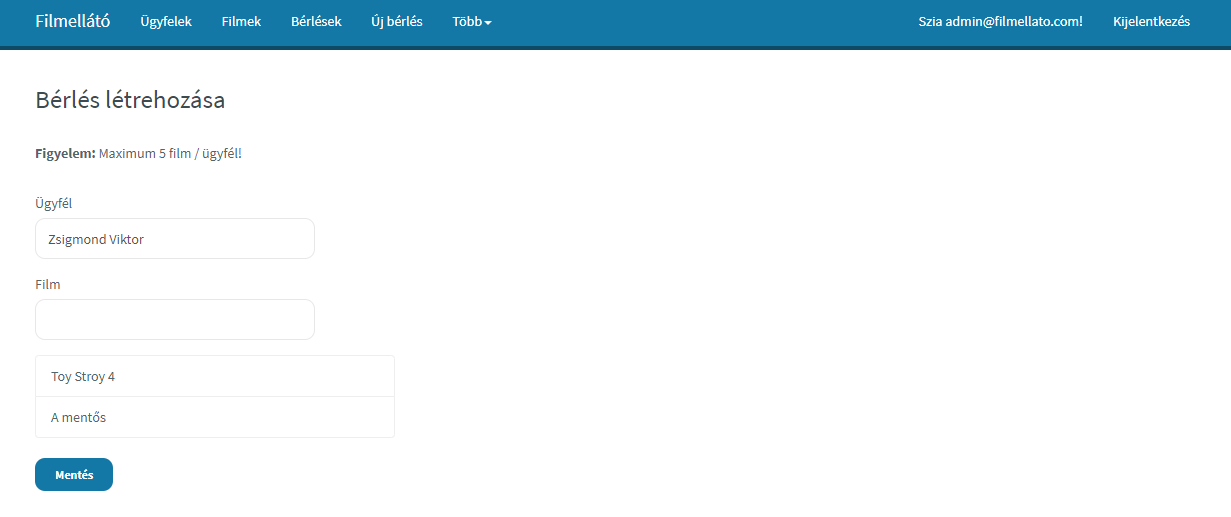
22. ábra: „Dolgozók listája” menüpont

### Dolgozó törlése

Dolgozót törölni csak a tulajdonosi (admin) felhasználói fiókból van lehetőség a „Több” legördülő menüben a „Dolgozók listája” menüpont alatt, a dolgozó neve melletti „Törlés” gombra kattintva.

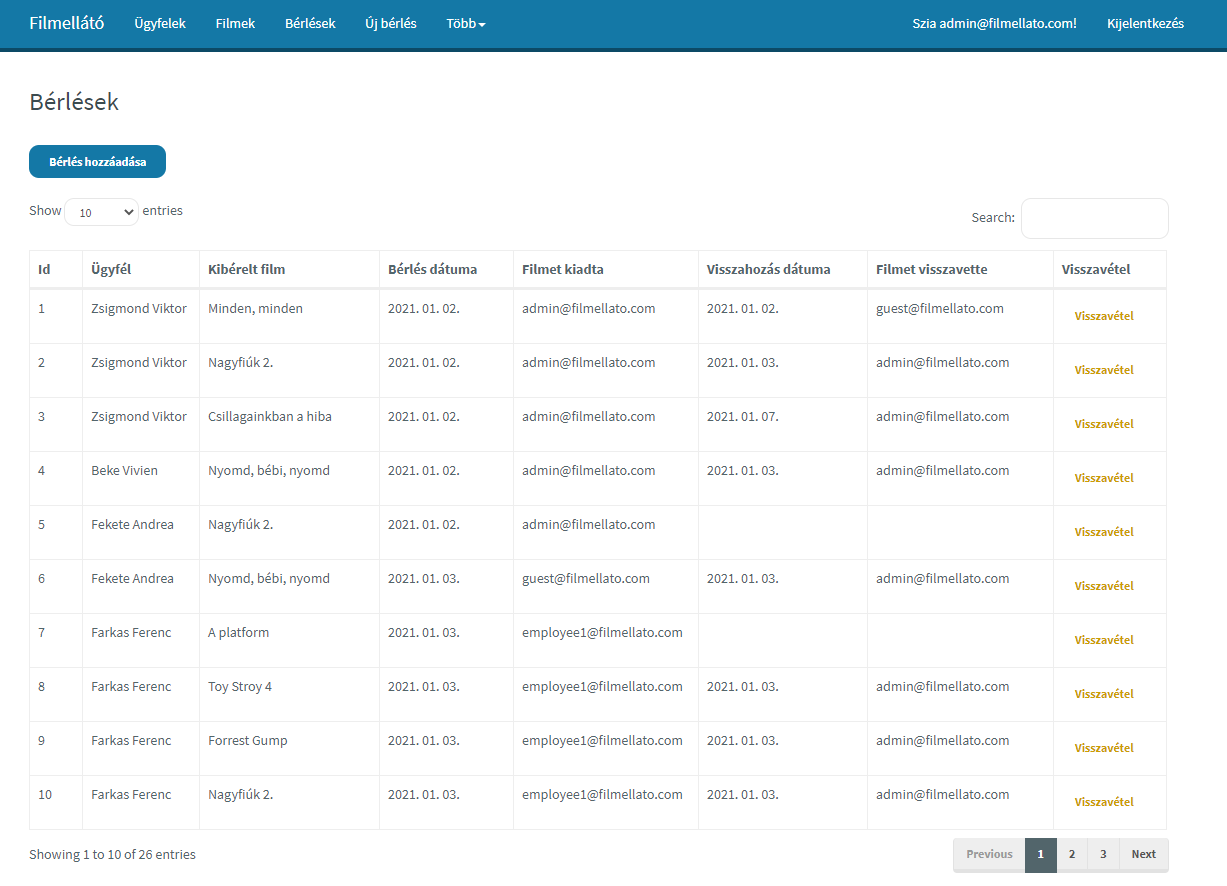
23. ábra: Dolgozó törlése

### Film bérbeadása

Filmet kiadni a dolgozói és a tulajdonosi (admin) felhasználói fiókból is lehet az „Új bérlés” menüpont alatt, amennyiben a keresett filmből van még elérhető mennyiség, a bérlő ügyfél nincs blokkolva és a bérlő ügyfélnél még nincs 5db film. Amennyiben valamelyik feltétel nem teljesül, a rendszer nem fogja engedni a bérlés létrehozását. Amennyiben valakinél 0db film van, akkor is maximum 5db-ot tudunk kiadni neki.

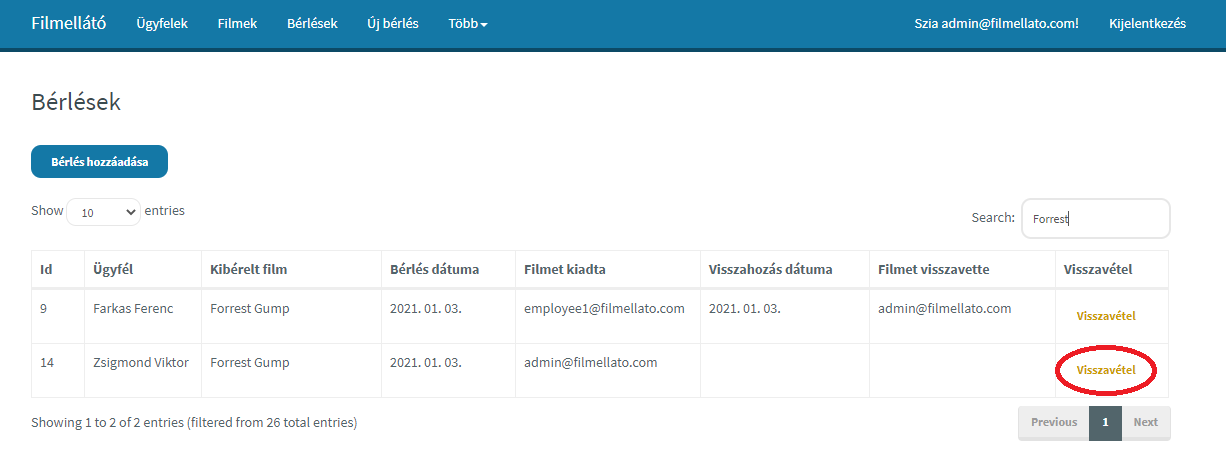
24. ábra: „Új bérlés” menüpont

### Kibérelt filmek megtekintése

A bérlések állapotát a dolgozói és a tulajdonosi (admin) felhasználói fiókból is meg lehet tekinteni a „Bérlések” menüpont alatt. Itt megtaláljuk a már visszahozott, valamint a még nem visszahozott bérlések adatait. Ki az ügyfél, melyik filmeket vette ki, mikor vette ki, ki adta ki a filmet, valamint, hogy mikor hozta azokat vissza és ki vette őket vissza.

25. ábra: „Bérlések” menüpont

### Kibérelt film visszavétele

Kibérelt filmet visszavenni a dolgozói és a tulajdonosi (admin) felhasználói fiókból is lehetőség van, amennyiben a filmet még nem hozták vissza. Ezt a „Bérlések” menüpont alatt a bérlés adatainak jobb oldalán lévő „Visszavétel” gomb megnyomásával érjük el.

26. ábra: Bérlés visszavétele

# Fejlesztői dokumentáció

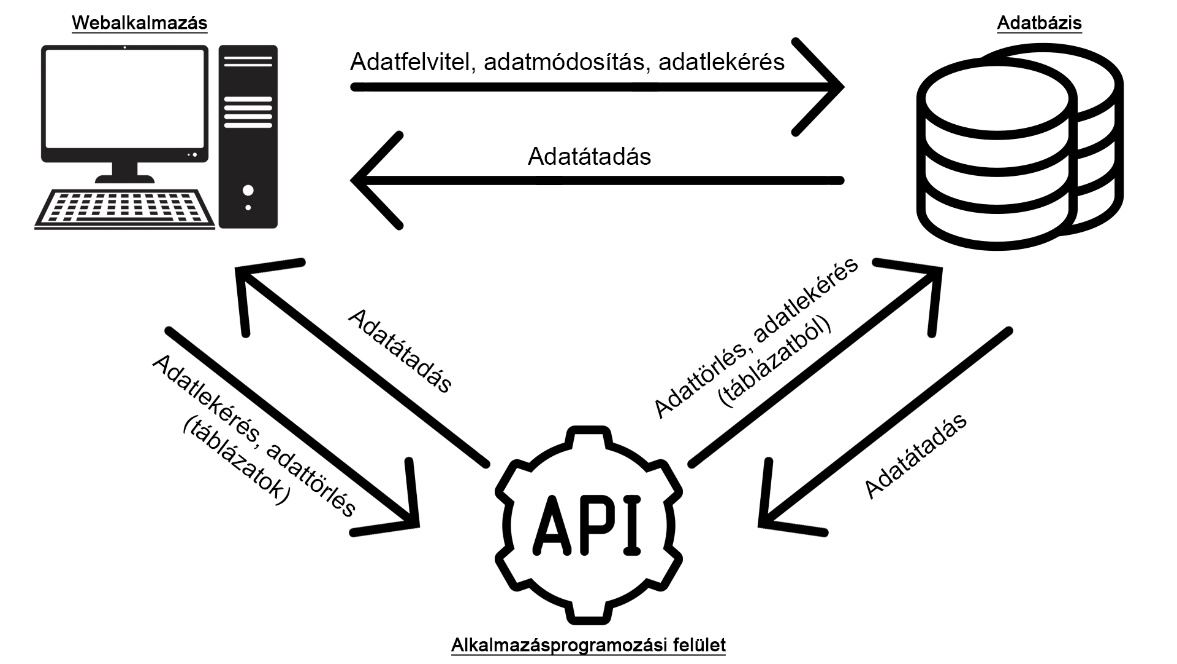
## Témamegjelölés

Egy „régimódi”, retró videótéka mindennapi tevékenységeit menedzselő weboldal elkészítése volt a cél, CodeFirst szemlélet követésével. Ügyfelek regisztrálása, módosítása, törlése, filmek felvitele, módosítása, törlése, dolgozók regisztrálása, törlése, valamint a bérlések nyomonkövetése tartozik a gyakori feladatok közé.

Témaválasztásom alapjául szolgált a filmek iránti szeretetem, továbbá az ASP.NET Framework alapú weboldalak irányába tanúsított érdeklődésem.

## Rendszerterv

A rendszer alapvetően egy felületből áll, ami nem más, mint maga a webalkalmazás. Ez kommunikál az adatbázissal, amiben a dolgozók, filmek, ügyfelek, bérlések adatait tároljuk.

A különböző menüpontok alatt („Ügyfelek”, „Filmek”, „Dolgozók”, „Bérlések”) találkozhatunk táblázatokkal, melyekben kereshetünk szöveg alapján, illetve rendezhetünk oszlopok alapján (a felhasznált eszközre a „Felhasznált eszközök, technológiák” pont alatt részletesebben is kitérek). Ezen táblázatok tartalma, illetve a kinézete nem a szerveren, hanem a kliens (felhasználó) számítógépén generálódik, hogy a keresés, rendezés ne a szerver erőforrásait használja, valamint, hogy gyorsabb legyen. Ehhez az információkat egy-egy API-n () keresztül éri el, így azoknak nincs folyamatos, közvetlen kapcsolatuk az adatbázissal.

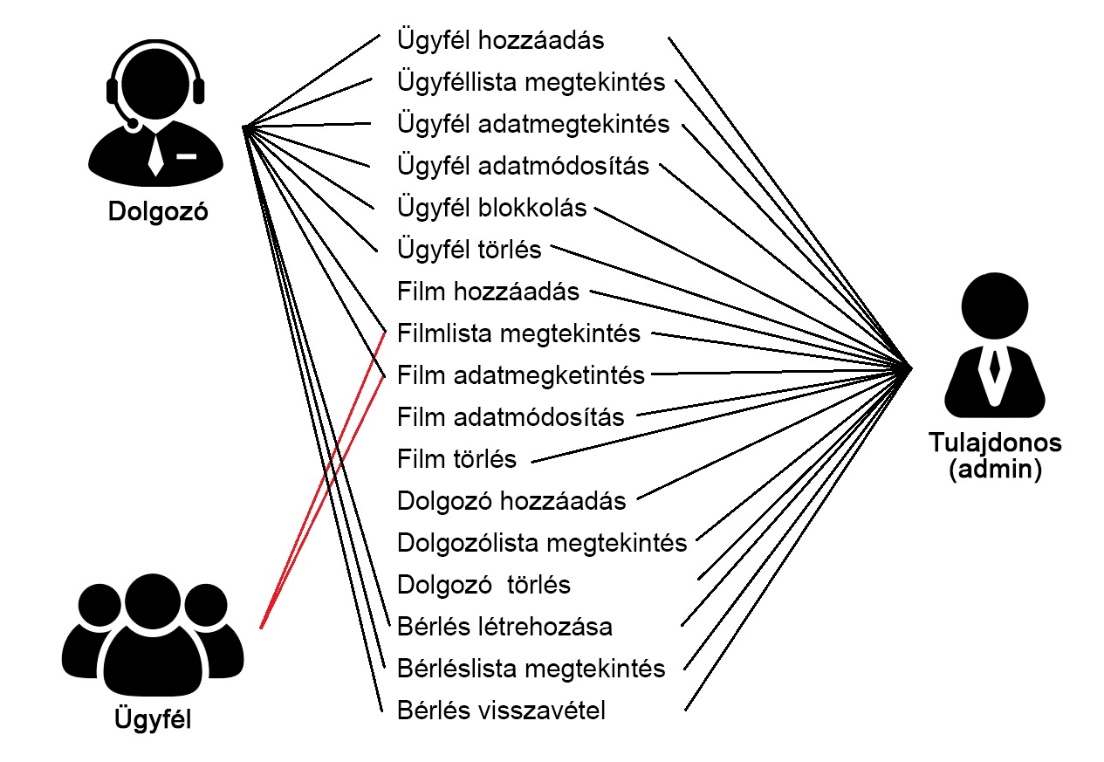
27. ábra: Rendszerterv

## Jogosultsági szintek

Alapvetően 3 különböző jogosultság van a webalkalmazásban: látogató, dolgozó, tulajdonos (admin).

A látogató nem tud belépni, nem tud adatot módosítani, viszont lát bizonyos adatokat. A filmeken, illetve a filmek adatain kívül látja még a „Rólunk” és a „Kapcsolat” oldalakat. Nem látja az ügyfeleket, nem látja a bérléseket és nem látja természetesen a dolgozókat sem. Betekintést nyerhet az aktuális filmkínálatba, a főoldalon látja a kiemelt ajánlatokat, valamint a filmek listáját táblázaton keresztül éri el, ahol kereshet szöveg alapján (cím / film fajtája), ezen kívül a filmeket tudja rendezni az oszlopok mentén (cím / bemutatás dátuma / film hossza percben megadva / film fajtája).

A dolgozó természetesen be tud lépni, így ér el plusz funkciókat. Egyrészt a dolgozó a filmekről is több információt tud (ID / elérhető darabszám). Ezen felül hozzáfér az ügyfelek adataihoz, valamint tud ügyfelet módosítani, törölni és létrehozni is. Látja a bérléseket, ezáltal tud filmet visszavenni, valamint tud új bérlést létrehozni is.

A tulajdonos (admin) természetesen szintén be tud lépni, ezáltal ér el még több funkciót. A dolgozó jogosultságain felül tud filmet felvenni, módosítani, illetve törölni. Ezen kívül még tud új dolgozót felvenni, és törölni is.

28. ábra: Jogosultsági szintek

## Adatbázis terv

Az adatbázis tervezésénél a fő szempont az volt, hogy az megfeleljen minden alapelvnek, valamint, hogy a való életben létező entitások (ügyfelek, dolgozók, tulajdonos, filmek, bérlések, stb…) a lehető legegyértelműbben és legegyszerűbben le legyenek modellezve. Ennek megfelelően a webalkalmazás az alábbi táblákat használja, melyek között természetesen nem szerepelnek az ASP.NET Framework által alapértelmezetten létrehozott táblák (ez alól kivétel a AspNetUsers tábla, mert azt szervesen használja az alkalmazás):

* dbo.Customers (ügyfelek)
* dbo.MembershipTypes (ügyféltagságok fajtái)
* dbo.Movies (filmek)
* dbo.Genres (filmtípusok)
* dbo.Rentals (bérlések és azok adatai)
* dbo.AspNetUsers (a regisztrált felhasználók (1 tulajdonos, és a dolgozók))

Az oldal látogatói, akik nem tudnak bejelentkezni, csak lekérdezni tudnak az adatbázisból a filmek listájának, valamint azok adatainak megtekintésével és azok keresésével. Emiatt számukra nem lett létrehozva tábla, ugyanis nem szükséges semmilyen adatot eltárolni velük kapcsolatban. Ezen felül természetesen jelen vannak az ASP.NET Framework által alapértelmezetten létrehozott táblák is:

* dbo.\_MigrationHistory (a migrációk előzményei, mely a CodeFirst szemlélet követése miatt vannak jelen)
* dbo.AspNetRoles (a különböző regisztrált felhasználói típusok, jelen projektben csak 1-et tartalmaz, a CanManageMovies nevűt, ami az adminé, mivel csak az tud filmeket kezelni)
* dbo.AspNetUserRoles (az előző táblában található típus hozzárendelése az admin fiókjához)
* dbo.AspNetUserClaims (jelen projektben nincs használatban)
* dbo.AspNetUserLogins (jelen projektben nincs használatban)

A táblák létrehozásánál természetesen figyelembe volt véve, hogy csak annyi, és olyan információ kerüljön eltárolásra, amelyek ténylegesen feltétlenül szükségesek a webalkalmazás működése szempontjából.

### Az ügyfelekről eltárolandó információk

A következő információk eltárolásával egyértelműen azonosíthatóvá válik egy-egy ügyfél:

* Az ügyfél neve
* Tagságának típusa
* Születési dátuma
* E-mail címe
* Telefonszáma
* Lakcíme
* Személyigazolvány száma
* Kép az ügyfélről

### A dolgozókról eltárolandó információk

Ha feltételezzük, hogy a dolgozókkal munkaszerződést is kötnek (ami kötelező), a következő információk eltárolásával egyértelműen azonosíthatóvá válik egy-egy dolgozó:

* A dolgozó e-mail címe
* A dolgozó telefonszáma
* A dolgozó felhasználóneve (jelen projektben az e-mail cím)
* Személyigazolvány száma

### A filmekről eltárolandó információk

A következő információk eltárolásával egyértelműen azonosíthatóvá válik egy-egy film:

* A film címe
* Bemutatásának dátuma
* Típusa
* Rendezője
* Hossza percben megadva
* Rövid leírása
* Borítóképe

### A kiadott filmekről (bérlések) eltárolandó információk

A következő információk eltárolásával egyértelműen azonosíthatóvá válik egy-egy bérlés, valamint azok valamilyen emberi hiba esetén (rossz DVD a tokban, stb…) egyértelműen visszakereshetőek:

* A film kiadásának dátuma
* A film visszahozásának dátuma (ha még nem hozták vissza, NULL)
* Az ügyfél ID-ja, aki kivette a filmet
* A film ID-ja, amit kivett
* A dolgozó e-mail címe, aki kiadta a filmet
* A dolgozó e-mail címe, aki visszavette a címet (ha még nem hozták vissza, NULL)

Természetesen a táblák tervezésénél figyelembe lettek véve a normálformák. Ennek mentén lettek létrehozva a tábláim, melyeket már fentebb felsoroltam, a következő módon:

### Customers (ügyfelek) tábla:

|  |  |
| --- | --- |
| **Id** | Egész (integer), a tábla elsődleges kulcsa, idegen kulcsa a **Rentals** tábla **Customer\_Id** mezőjének az ügyfél egyedi azonosítója, nem lehet NULL |
| Name | Szöveg (varchar), az ügyfél neve, nem lehet NULL |
| IsSubscribedToNewsletter | Logikai (boolean), az ügyfél feliratkozott-e a hírlevél szolgáltatásra, nem lehet NULL |
| MembershipTypeId | Egész, idegen kulcsa a **MembershipTypes** tábla **Id** mezőjének, megadja, hogy az ügyfél milyen tagságra fizetett elő, nem lehet NULL |
| BirthDate | Dátum (DateTime), az ügyfél születési ideje |
| IsBlocked | Logikai (boolean), az ügyfél blokkolva lett-e, nem lehet NULL |
| Email | Szöveg (varchar), az ügyfél e-mail címe, nem lehet NULL |
| PhoneNumber | Szöveg (varchar), az ügyfél telefonszáma, nem lehet NULL |
| Address | Szöveg (varchar), az ügyfél lakcíme, nem lehet NULL |
| IdentityCard | Szöveg (varchar), az ügyfél személyigazolványszáma, nem lehet NULL |
| NumberOfCurrentlyRentedMovies | Egész (integer), hány kibérelt film van épp az ügyfélnél, nem lehet NULL |
| ImagePath | Szöveg (varchar), az ügyfél feltöltött képének elérési útja |

### MembershipTypes (ügyféltagságok fajtái) tábla:

|  |  |
| --- | --- |
| **Id** | Egész (tiny integer), a tábla elsődleges kulcsa, a **Customers** tábla **MembershipTypeId** mezőjére mutat, megadja az ügyfél tagságát, nem lehet NULL |
| SignUpFee | Egész (small integer), a tagság díja forintban megadva, nem lehet NULL |
| Duration | Egész (tiny integer), a tagság időtartama hónapban megadva, nem lehet NULL |
| DiscountRate | Egész (tiny integer), az adott tagsággal mennyi kedvezményre jogosult az ügyfél százalékban megadva, nem lehet NULL |
| MembershipName | Szöveg (varchar), a tagság megnevezése |

### Movies (filmek) tábla:

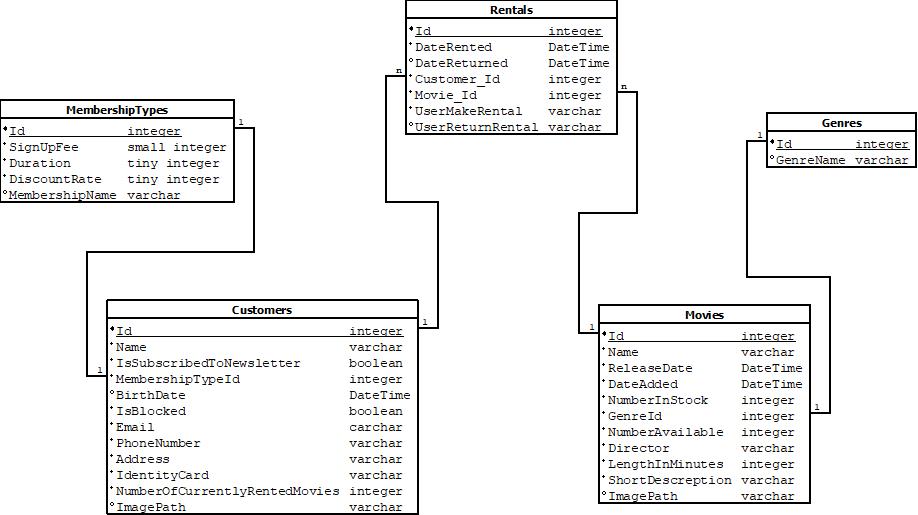
|  |  |
| --- | --- |
| **Id** | Egész (integer), a tábla elsődleges kulcsa, idegen kulcsa a **Rentals** tábla **Movie\_Id** mezőjének, a film egyedi azonosítója, nem lehet NULL |
| Name | Szöveg (varchar), a film címe, nem lehet NULL |
| ReleaseDate | Dátum (DateTime), a film bemutatásának dátuma, nem lehet NULL |
| DateAdded | Dátum (DateTime), a film rendszerbe történő feltöltésének ideje, nem lehet NULL |
| NumberInStock | Egész (integer), hány darab van összesen a filmből, nem lehet NULL |
| GenreId | Egész (integer), idegen kulcsa a **Genres** tábla **Id** mezőjének, megadja a film típusát, nem lehet NULL |
| NumberAvailable | Egész (integer), hány darab elérhető még a filmből (bérelhető), nem lehet NULL |
| Director | Szöveg (varchar), a film rendezője, nem lehet NULL |
| LengthInMinutes | Egész (integer), a film hossza percben megadva, nem lehet NULL |
| ShortDescription | Szöveg (varchar), a film rövid leírása / tartalma, nem lehet NULL |
| ImagePath | Szöveg (varchar), a film feltöltött borítóképének elérési útja |

### Genres (filmtípusok) tábla:

|  |  |
| --- | --- |
| **Id** | Egész (integer), a tábla elsődleges kulcsa, a **Movies** tábla **GenreId** mezőjére mutat, megadja a film típusát, nem lehet NULL |
| GenreName | Szöveg (varchar), a film típusának neve |

### Rentals (filmbérlések) tábla:

|  |  |
| --- | --- |
| **Id** | Egész (integer), a tábla elsődleges kulcsa, a bérlés egyedi azonosítója, nem lehet NULL |
| DateRented | Dátum (DateTime), a bérlés létrejöttének ideje, nem lehet NULL |
| DateReturned | Dátum (DateTime), a film visszahozásának ideje |
| Customer\_Id | Egész (integer), idegen kulcsa a **Customers tábla Id** mezőjének, megadja a bérlő ügyfelet, nem lehet NULL |
| Movie\_Id | Egész (integer), idegen kulcsa a **Movies** tábla **Id** mezőjének, megadja a bérelt filmet, nem lehet NULL |
| UserMakeRental | Szöveg (varchar), a bérlést létrehozó dolgozó felhasználóneve (e-mail címe), nem lehet NULL |
| UserReturnRental | Szöveg (varchar), a filmet visszavételező dolgozó felhasználóneve (e-mail címe) |

Természetesen, hogy az adatbázis és annak táblái, valamint kapcsolataik átláthatóbbak legyenek, elkészült hozzá az egyed-kapcsolat diagram:

29. ábra: Adatbázis egyed-kapcsolat diagram

A MembershipTypes és a Genres táblák migrációk segítségével töltődnek fel adatokkal, ezzel elérve, hogy a webalkalmazás első indítása során is alkalmazhatóak legyenek. A többi tábla feltöltéséhez a webalkalmazásban található űrlapokat kell használni (Ügyfél hozzáadása, Film hozzáadása (csak tulajdonos (admin)), Új bérlés hozzáadása).

Mivel a webalkalmazás CodeFirst szemlélet segítségével készült el, így a tervezés után készítettem el a projekthez szükséges osztályokat (modelleket), melyekből migráció segítségével készültek el az adatbázis táblák. Ezek a modellek a projekt mappáján belül a Models mappában találhatóak, a migrációk pedig a Migrations mappában.

## Felhasznált eszközök, technológiák

### C# programozási nyelv

A C# egy, a Microsoft által fejlesztett, .NET keretrendszer alá tartozó programozási nyelv. Lényege, hogy bármely .NET keretrendszert telepített eszközön futtatható programokat tudunk írni segítségével. Továbbá fontos, hogy a C# egy erősen típusos, OOP (objektumorientált programozás) nyelv, ami azt jelenti, hogy az IDE (fejlesztői környezet – jelen esetben Visual Studio), valamint a nyelv segít nekünk abban, hogy a programunk megfeleljen az OOP alapelveknek.

Projektem teljes üzleti logikáját ezen nyelv segítségével készítettem el. Választásom azért erre a nyelvre esett, mert középiskolai tanulmányaim óta C#-ot tanulok, illetve a programozás alapjait is ezen nyelven keresztül sajátítottam el, vagyis ebben van a legtöbb tapasztalatom, valamint „rutinosnak” érzem magam a Visual Studio fejlesztői környezetben.

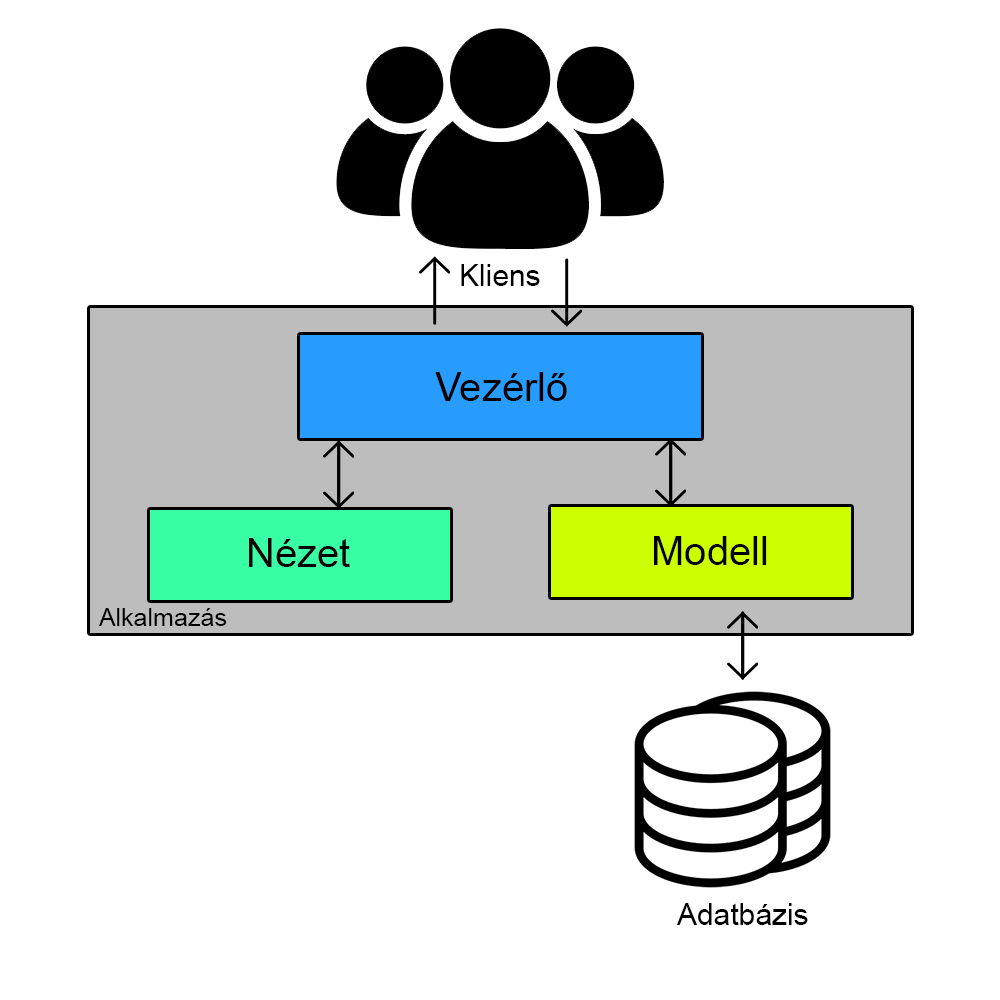
### ASP.NET MVC keretrendszer

A projektem tervezésekor mindenképpen szerettem volna egy webes alkalmazást elkészíteni, valamint a C# nyelv használati lehetősége is prioritást élvezett. Továbbá a szoftverfejlesztő képzés keretein belül találkoztunk a webalkalmazások MVC (Model – View - Controller) alapú felépítésével, így abban is volt már némi tapasztalatom. Eme három dolog mentén választottam az ASP.NET MVC keretrendszert webalkalmazásom elkészítéséhez.

Maga a .NET, mint azt már az előző pontban (C# programozási nyelv) is kifejtettem, egy a Microsoft által létrehozott keretrendszer. Az ASP (Active Server Page) pedig egy dinamikus weboldalak elkészítését lehetővé tévő, osztályok és komponensek együttese, melyet bármely .NET keretrendszer által támogatott programozási nyelvvel használhatunk.

Ebbe az ASP.NET csomagba tartozik bele az ASP.NET MVC is. Az ebben megírt webalkalmazások a már említett MVC tervezési minta alapján készülnek el, mely magyarra fordítva a Modell – Nézet – Vezérlő. Ennek lényege, hogy az alkalmazás komponenseit ezen három részre osztja szét, melynek elsődleges célja, hogy a felhasználói felület megjelenítéséért felelős elemeket teljesen elválasszuk a weboldal működését szolgáló elemektől. Ez a fajta szétválasztás nagyban megkönnyíti a tervezést, fejlesztést, valamint a hibajavítást is.

* **M**odel (Modell) – Az adatbázisban található tábláinknak megfelelő C# osztályok, melyekben az adatbázis rekordjait tudjuk tárolni, ezáltal azokat objektumként használni. Mondhatni az adatbázis adatait kezeli.
* **V**iew (Nézet) – Tulajdonképpen ezek felelősek a felhasználói élményért, az üzleti logika grafikai megjelenítéséért. Ezeken keresztül adunk, valamint fogadunk információt a felhasználó számára.
* **C**ontroller (Vezérlő) – Alapvetően a felhasználói adatbevitel feldolgozásáért, valamint a megfelelő Nézetek betöltéséért felel. Ezeken keresztül érjük el a Modelleket, vagyis közvetetten az adatbázist.

Az ASP.NET MVC 2010 óta a Visual Studio részét képezi, vagyis egy ilyen projekt létrehozása rendkívül egyszerű, valamint tökéletesen használható együtt az Entity Framework-el.

30. ábra: Modell – Nézet - Vezérlő

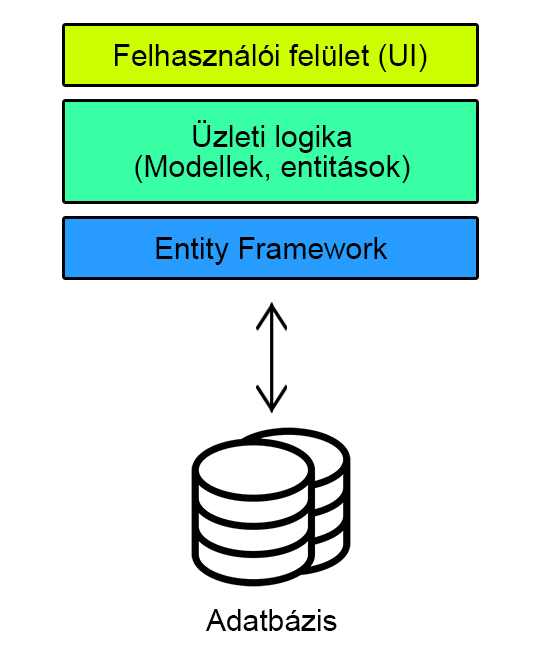
### Entity Framework

A webalkalmazásom adatbázissal kapcsolatos feladatait az Entity Framework-kel valósítottam meg. Választásom azért esett erre, mert maga is egy Microsoft által kifejlesztett rendszer, így tökéletesen működik a projektem többi összetevőjével. Lényegében ez egy ORM (Object-Relational Mapper), avagy objektum-relációs leképező rendszer. Legfontosabb feladata, hogy létrehozza az adatainkat megvalósító objektumokat az adatbázis tartalma alapján, valamint az azok között lévő kapcsolatot is, ezáltal lehetővé téve az adatbázis műveletek elvégzését C# objektumokon keresztül.

Az Entity Framework segítségével projektünk elkészítését két irányból is megkezdhetjük: „Database First”, vagy „Code First” szemlélettel.

A „Database First” logika alapján először megtervezzük az adatbázisunkat, létrehozzuk a tábláinkat, a közöttük lévő kapcsolatokat, majd az Entity Framework ezek alapján hozza létre a Modelleket, amelyek az adatainkat reprezentálják. Nevét pont emiatt kapta, hiszen először magával az adatbázissal foglalkozunk, majd utána hozzuk létre a Modelleket.

A „Code First” megközelítéssel először létrehozzuk a Modelljeinket. Később ezek lesznek majd az adatbázisunk táblái, melyet az Entity Framework a Modellek alapján automatikusan legenerál nekünk először migráció formájában, természetesen a táblák közötti kapcsolatokkal együtt. Amennyiben a migrációs fájlban mindent rendben találunk, azt lefuttatva tudjuk végrehajtani a módosításokat ténylegesen az adatbázison is. Én személy szerint a „Code First” szemléletet követtem a projektem elkészítése során, egyszerűen csak ezt éreztem kényelmesebbnek, kézenfekvőbbnek.

További hasznos tulajdonsága az Entity Framework-nek, hogy Visual Studio-val használható LINQ-s lekérdezéseinket automatikusan SQL lekérdezésekké tudja alakítani, ezáltal használata rendkívül egyszerűvé válik.

31. ábra: Entity Framework

### 

### Internet Information Services (IIS)

Az Internet Information Services egy, a Microsoft által fejlesztett webszerver, mely képes futni bármely Windows eszközön, valamint az ASP.NET keretrendszerben írt webalkalmazások futtatására is képes az eddig felsorolt technológiákkal. Az IIS támogatja természetesen a HTTP, HTTPS, FTP, FSTP, SMTP és NNTP protokollokat. Választásom azért az IIS-re esett, mert ez minden általam felhasznált technológiával képes együtt műküdni.

### jQuery

A jQuery egy, a leggyakrabban használt JavaScript könyvtár. Feladata, hogy a JavaScript kódot minél inkább le tudjuk választani a HTML kódról, valamint, hogy kényelmes kommunikációt biztosítson a weblap elemeivel. Záródolgozatomban használata a DataTables és a BootBox pluginoknál jött elő, ahol AJAX hívások megvalósítására használtam.

### Bootstrap

A Bootstrap lényegében egy nyílt forráskódú front-end keretrendszer. Azért jött létre, hogy leegszerüsítse a dinamikus weboldalak elkészítését. Használatával egy webalkalmazás front-end oldala mindenféle eszközön képes lesz normálisan megjelenni, nem fognak kilógni részek sehol. Webalkalmazásom fejlesztése során fontos volt, hogy az szépen jelenjen meg mobil eszközökön is, hiszen a weboldal alapvetően bárki által látogatható, megtekinthető.

# Tesztelési dokumentáció

## Webalkalmazás adatbázisműveletek tesztelése

Az adatbázisműveletek tesztelését manuálisan végeztem el. Lehetőségemhez mérten igyekeztem minden olyan helyzetre, lehetőségre tesztelést végezni, ami a webalkalmazás használata során előjöhet, melyek főként a CRUD (**C**reate, **R**ead, **U**pdate, **D**elete / Létrehozás, Olvasás / Lekérdezés, Módosítás, Törlés) műveletek.

Ezeket a teszteket a webalkalmazáson keresztül is elvégeztem, hiszen azon keresztül is tudjuk módosítani az adatbázist (filmek, ügyfelek, bérlések, dolgozók hozzáadása, adatainak megtekintése, módosítása, valamint törlése).

## Felhasználói felület (GUI) tesztelése

A felhasználói felület (GUI) tesztelését manuálisan végeztem el. Igyekeztem minden olyan útvonalat bejárni, ami rendeltetés, valamint nem rendeltetésszerű használat mellett előjöhet. A webalkalmazás reszponzív megjelenését a Firefox Developer Edition böngésző segítségével végeztem, azon belül is a „Responsive Design Mode” használatával.

# Összegzés

## Amit sikerült megvalósítani

A projektemmel szemben támasztott elképzeléseimet, valamint ötleteimet mind egytől-egyik sikerült megvalósítanom, implementálnom, ami a filmeket, ügyfeleket, bérléseket, dolgozókat érintő funkciókat jelenti.

## Fejlesztési lehetőségek

Természetesen projektemet számos ponton tovább lehetne fejleszteni. Egyrészről a használaton sokat javítana, hogyha a tulajdonos (admin) nem csak normál dolgozót, hanem másik tulajdonost (admint) is tudna regisztrálni, hogy ne mindig egy személyhez legyen kötve az összes funkció, mely a filmek menedzselésével kapcsolatos.

Továbbá az előző iránynál maradva be lehetne vezetni egy dolgozói nyilvántartó rendszert is, mely feljegyezné, hogy melyik dolgozó melyik napon, pontosan mikor jelentkezett be (ért be munkába), illetve, hogy mikor jelentkezett ki (indult haza). Ezen keresztül a szabadságolást, a szabadságok kérését, valamint a betegszabadságokat is nyilván lehetne tartani.

A webalkalmazásnak lehetne készíteni akár egy letölthető mobilapplikációt is, melyen keresztül a látogatók megtekinthetnék a filmkínálatot, illetve akár a dolgozók is tudnák intézni az ügyeiket. Továbbá, hogy az applikáció igazi értelmet nyerjen, be lehetne vezetni valamilyen „streaming” szolgáltatást is, ahol az előfizetők megtekinthetnék az elérhető filmeket online.

## Személyes tapasztalatok

A záródolgozatom tervezésénél kitűzött céljaim mindegyikét sikerült megvalósítanom, ezzel pedig teljesítettem a magammal, valamint a projektemmel szemben támasztott saját elvárásaimat.

A szoftverfejlesztő OKJ során szerzett tudásomat remekül tudtam kamatoztatni, főleg azt, melyet a C# fejlesztés órákon szereztem. Ez a projekt rendkívül jó lehetőséget adott arra, hogy fejlesszem, tágítsam a webfejlesztéssel, főként az ASP.NET-el, valamint az MVC alapú webalkalmazásokkal kapcsolatos ismereteimet, tudásomat. A záródolgozatom készítése során megtapasztaltam az előre tervezés fontosságát, valamint, hogy hogyan is lehet egy viszonylag komplex feladatot több, jól átlátható, tagolt részegységre bontani.

Véleményem szerint az elkészült projektem egy jól átlátható, felhasználóbarát, a mindennapi élet során használható, biztonságos, valamint az alapvető követelményeknek megfelelő szoftver.

## Felhasznált források

* <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>
* <https://en.wikipedia.org/wiki/ASP.NET>
* <https://en.wikipedia.org/wiki/ASP.NET_MVC>
* <https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91>
* <https://hu.wikipedia.org/wiki/Internet_Information_Services>
* <https://hu.wikipedia.org/wiki/JQuery>
* <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>
* <https://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-9-12-evfolyam/alapok-a-nyelv-tortenete-jellemzoi-a-kornyezet-hasznalata/c-programozasi-nyelv>
* <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>
* <https://www.udemy.com/course/the-complete-aspnet-mvc-5-course/>
* <https://docs.microsoft.com/en-us/iis/extensions/introduction-to-iis-express/iis-express-overview>
* <https://visualstudio.microsoft.com>
* <https://dotnet.microsoft.com/apps/aspnet/mvc>
* <https://portableapps.com/apps/office/dia_portable>
* <https://twitter.github.io/typeahead.js/examples/>
* <https://github.com/CodeSeven/toastr>
* <https://developer.mozilla.org/en-US/docs/Web>
* <https://automapper.org/>
* <https://elmah.github.io>
* <https://modernizr.com>
* <https://datatables.net>
* <https://getbootstrap.com>
* <http://bootboxjs.com>
* <https://www.photoshop.com/>
* <https://jquery.com/>
* <https://github.com/>
* <https://stackoverflow.com/>
* <https://git-scm.com/>

## Köszönetnyilvánítás

Köszönettel tartozom konzulensemnek, Répás Csabának, aki minden C# nyelvvel kapcsolatos felmerülő kérdésemre választ adott, valamint segített a megfelelő, minőségi megoldások megtalálásában.

Továbbá köszönettel tartozom Rostagni Csabának, akitől elsajátíthattam az adatbázisokkal, valamint az MVC alapú webalkalmazásokkal kapcsolatos tudásomat, ismereteimet.

## Ábrajegyzék