

Szegedi Szakképzési Centrum Vasvári Pál Gazdasági és Informatikai
Szakgimnáziuma

54 213 05 számú Szoftverfejlesztő szakképesítés záródolgozata

cinemaArk

mozi weboldal és adminisztrációs rendszer

Készítette:

Pál Máté

Szeged

2021

Hallgatói Nyilatkozat

Alulírott

Pál Máté

a Szegedi Szakképzési Centrum Vasvári Pál Gazdasági és Informatikai Szakgimnáziuma
hallgatója kijelentem, hogy a
cinemaArk, mozi weboldal és adminisztrációs rendszer
című záródolgozat a saját munkám.

Kelt: Szeged, 2021. március 29.

Pál Máté

TARTALOMJEGYZÉK

1. BEVEZETŐ	1
1.1 TÉMAVÁLASZTÁS.....	1
1.2 MEGRENDELŐI IGÉNYEK	2
1.3 FEJLESZTÉS SORÁN HASZNÁLT PROGRAMOK	2
2. RENDSZERTERV	3
2.1 ADATBÁZIS TÁBLÁK	3
2.1.1 Táblák közötti kapcsolatok.....	4
3. A WEBOLDAL	6
3.1 HONLAP FELÉPÍTÉSE	7
3.1.1 Menüsáv elemei.....	7
4. AZ ASZTALI ALKALMAZÁS	10
4.1 MENÜSÁV ELEMEL.....	10
4.2 OBJEKTUM ORIENTÁLT PROGRAMOZÁS (OOP)	12
4.3 MODELL-NÉZET-PREZENTER (MVP).....	14
4.4 LINQ (LANGUAGE-INTEGRATED QUERY)	15
4.5 STÁTUSZ	16
5. TESZTELÉS	17
6. FEJLESZTÉSI LEHETŐSÉGEK.....	18
6.1 A WEBLAP TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEI.....	18
6.2 AZ ASZTALI ALKALMAZÁS TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEI	20
7. HASZNÁLATI ÚTMUTATÓ.....	21
7.1 BEVEZETÉS	21
7.2 TELEPÍTÉS.....	21
7.3 BEJELENTKEZÉS.....	21
7.4 KEZDŐLAP	22
7.5 MENÜSOR HASZNÁLATA	22
8. ÖSSZEGZÉS	25
9. FELHASZNÁLT IRODALOM.....	26
9.1 FORRÁSOK	27
10. MELLÉKLETEK.....	28

1. BEVEZETŐ

A cinemaArk egy fiktív mozi weboldala, mely lehetőséget nyújt az előadások online jegyfoglalására, ezzel felgyorsítva a jegyvásárlás folyamatát a fizikai kasszáknál azáltal, hogy az azonosítást követően már nem kerül sor további adminisztrációs munkákra, így felgyorsul a jegyvásárlás és hamarabb jutnak el a mozit látogatók az általuk előre kiválasztott helyekre. További funkciói közé tartozik a látogatók tájékoztatása a hamarosan vászonra kerülő filmekről, illetve a regisztrált felhasználók könnyű kapcsolattartása a cég munkatársaival.

A cég neve az angol cinema(magyarul: mozi) és ark(magyarul bárka szóból) ered. A cinema adott volt, sok cég használja fel előszeretettel az érthetőség miatt, hogy mi a cég profilja, a bárka pedig egyfajta egyedi jelzőként szolgál, ami segítségével megjegyezhetik könnyen a céget. Háttér jelentése tartalmazza azt is, hogy aki a cinemaArk-al tart, izgalmas vizekre evezhet és új élményekben lehet része.

A weboldalhoz tartozik egy adminisztrációs asztali alkalmazás, melyben lehetőség van az adatok módosítására a weboldalon egy adatbázis segítségével. Ennek részletes ismertetésére a további oldalakon sor kerül.

1.1 Témaválasztás

Témaválasztásomnak 3 oka van:

Az első, hogy a jövőben szeretnék weboldalak készítésével foglalkozni és egy viszonylag nehezebb, validációkkal teli weboldal elkészítését akartam megvalósítani. A validációk olyan feltételek, algoritmusok, melyek elengedhetetlenek a megfelelő, életszerű működéshez és egy mozi weboldala tökéletes választásnak bizonyult a jegyfoglalás bonyolultságának köszönhetően. Ezzel azt értem el, hogy a későbbiekben az egyszerűbb weboldalak akadályait könnyebben fogom átlépni.

A második, hogy megoldást nyújt ebben a felgyorsult, sokszor túlterhelt világban a moziba vágyóknak, ha a weboldal által nyújtott szolgáltatások megfelelően vannak integrálva a tényleges, cég által kialakított létesítményben. Egy egyszerű példát hozva a kasszáknál elérhető a látogatóknak egy külön sor, melynél az előre foglalt jegyeket lehet hitelesíteni rendkívül gyorsan. Hisz ki az, aki szeret lemaradni az előadás első perceiről?

Végül az utolsó ok, ami miatt az eddig taglalt oldal létrehozása mellett döntöttem, hogy jó referenciaként fog szolgálni a továbbiakban egy kis továbbfejlesztés után, melyet büszkén mutathatok meg másoknak, főleg, ha egy állásajánlatról van szó.

1.2 Megrendelői igények

Egy kis keresgélés után tiszta volt számomra, hogy milyen funkciókat kell beépítenem, mind a weboldalba, mind az asztali alkalmazásomba, hogy kielégítsem a megrendelők igényeit. Úgy gondolom ezek legfontosabb elemeit sikeresen beépítettem a szakdolgozatomba és akár többször is felhasználhatóak lesznek bizonyos elemek.

Az alábbi elemeket véltem fontosnak:

- A programban a weboldal felhasználóinak adatait lehessen tárolni az esetleges hibák fellépése miatt, hogy azokat könnyen módosítani lehessen.
- Új filmeket lehessen a weboldalra tenni a megfelelő jogosultsággal.
- kapcsolattartási lehetőség grafikus felületen a felhasználó és a rendszergazda között.
- A komplett jegyfoglalási rendszer.
- A foglalásokkal kapcsolatos hibák módosításának lehetősége.

1.3 Fejlesztés során használt programok

Maga a fejlesztés Windows 10 operációs rendszeren került elkészítésre.

A fejlesztés során különböző fejlesztői programokat használtam, ezek mind ingyenesen elérhetők az adott verzióban.

Weblapszerkesztés:

- Visual Studio Code (ingyenes community verzió, HTML CSS, PHP, JavaScript nyelvű kódok írásához)

Asztali alkalmazáshoz:

- Microsoft Visual Studio 2019 (C#)

Logó tervezéshez:

- Inkspace

Az adatbázishoz és minden más kapcsolat létrehozásához:

- XAMPP Control Panel v3.2.4

2. Rendszerterv

Az adatbázis tervezésekor fontos volt meghatároznom, hogy milyen típusú adatokat fogok tárolni, ezek az adatok hogyan fognak kapcsolódni egymáshoz és hogy mennyi táblára lesz szükségem. Ezeket mind megfelelően definiálnom kellett, oda figyelve arra, hogy csak az azonos értékű és különböző táblákban található mezőneveknél lépjen fel a redundancia. A redundancia azt jelenti, hogy valamilyen értéket többszörösen tárolunk az adatbázisban. 1-2 esetben azért lép fel a jelenség az adatbázisomban, mert kifejezetten szerettem volna, ha bizonyos táblák ténylegesen eltárolják ugyanazt az információt. De ezek többnyire index értékek és össze vannak kapcsolva egymással, a félreértések elkerülése végett.

A normalizálás módszerének szabályai is fellelhetők az adatbázisan. Például külön táblákba lettek elhelyezve a különböző fogalmakkal kapcsolatos adatok. Minden táblában meg lett határozva egy elsődleges kulcs, mely rendszerint egy index, ami megkönnyíti az adatok lekérdezését és sorszámozását.

2.1 Adatbázis táblák

A weblaphoz és az asztali alkalmazáshoz is ugyanazt az adatbázist használtam fel az átláthatóbb szerkezet és az egyszerűség végett. Ha több ügyfélnek is készítek szinte azonos projektet, akkor elengedhetetlen a megrendelők számától függő adatbázis létrehozása. Hisz valószínű, hogy az adatok tartalma részben eltér és biztonságosabb is, ha 1 céghez 1 külön adatbázis tartozik.

Az alábbi táblák találhatók meg az adatbázisban:

movies

- Ebben a táblában találhatók meg a műsoron lévő filmek és a hozzá kapcsolódó összes adat, ezek következők: cím, műfaj, hossz, értékelés, megjelenés dátuma, terem, ahol a film bemutatásra kerül és egy borító kép.

tickets

- itt vannak eltárolva a jegytípusok (ár és típus szerint), amikre jogosult lehet a látogató.

user

- A beregisztrált felhasználók adatai találhatóak meg ebben a táblában. Ez a tábla sűrűn fordul elő a különböző weblapok adatbázisában. Eltárol egy azonosítót (**id**), a felhasználónevet (**username**), a fiókhoz tartozó jelszót SHA512 titkosítással (**pwd**),

Hivatalos nevet (**name**), e-mail címet (email), lakcímet (**home_address**) és a regisztráció dátumát (**join_date**).

orders

- Az adott foglalással kapcsolatos információk. Fizetési mód (**payment_mode**), a megrendelés dátuma (**date**), a vásárló azonosító száma, mely kapcsolatban áll a user táblában található felhasználó id azonosítójával (**customer_id**).

items

- A foglaláshoz fűződő tételek. Minden, amit a jegy tartalmaz. (sorszám, az egyidejű rendelés darabszáma, jegytípus darabszáma, ár, megrendelés sorszáma, megrendelő azonosítója, előadás, sor, szék, hónap, nap, időpont).

contact

- Itt jelennek meg a felhasználók kérdései, problémái vagy amit közölni szeretne az adminisztrátorokkal, illetve a felhasználó azonosító száma (**message**, **name_id**).

lecture_time

- Eltárolja az előadás időpontokat.

seats

- Ülőhelyek kilistázva.

line

- A mozi termék sorai kilistázva.

A **days** és a **months** tábla létrehozása mellett azért döntöttem, hogy megkönnyítsem a kódban írt néhány kilistázásom.

2.1.1 Táblák közötti kapcsolatok

A táblák között fellelhetők kapcsolatok, mellyel megoldottam a redundanciát is, hisz bizonyos táblákban csak egy azonosítóval tároltam el adatokat. Például a felhasználó név több táblában is felbukkan **customer_id** néven, de a terem sor, és a szék száma is megjelenik a tételek között az **items** táblában.

Táblák közötti kapcsolatok típusai:

Egy-a-többhöz kapcsolat:

- Leggyakrabban használatos kapcsolati típus. Az egy-a-többhöz kapcsolatban az „A” tábla valamely rekordjához több rekord tartozhat a „B” táblából.

De a „B” tábla valamennyi rekordjához csak egy-egy rekord tartozhat az „A” táblába.

Több-a-többhöz kapcsolat:

- A tábla valamely rekordjához több rekord is tartozhat a „B” táblában, és a „B” tábla valamely rekordjához is több rekord tartozhat az „A” táblában.

Egy-az-egyhez kapcsolat:

- A tábla minden egyes rekordjához legfeljebb egy rekord tartozhat a B táblában, és a „B” tábla minden egyes rekordjához is csak legfeljebb egy rekord tartozhat az „A” táblában.

- Ez a fajta kapcsolat nem túl gyakran használatos, mert a legtöbb információ, amelyet ilyen módon írunk le, leírható egyetlen táblán belül is. Az egy-az-egyhez kapcsolat akkor lehet hasznos, ha egy sok rekordból álló táblát több kisebb, könnyebben kezelhető táblára kívánunk felosztani.

A táblák tartalmát a rendszergazda/adminisztrátor az asztali alkalmazásba bejelentkezés után képes a tartalmat szerkeszteni. Megtalálható az összes CRUD művelet a programban. Ez egy rövidítés mely a következőket jelenti:

C – create (létrehoz).

R – read (olvas).

U -update (módosít).

D – delete (töröl).

3. A weboldal

Egy egységes weblap kialakítása sok tényezőből áll, úgy gondolom ezt sikerült megvalósítanom. A weblap elkészítéséhez több webprogramozási nyelvet is használtam, elsőnek is a HTML-t (HyperText Markup Language) említem meg, mely segítségével hoztam létre a weboldalon található űrlapokat, táblázatokat, szerkezetet. Több HTML fájlt is tartalmaz a program. Ez azért van, hogy jól elkülönüljenek egymástól a különböző elemek és átláthatóbb legyen a kód. Másik jelentős oka, hogy ezeket a HTML tartalmakat meghívtam a PHP (Hypertext Preprocessor) fájljaiban és kellemetlen lett volna, ha 1 mindent tartalmazó HTML fájl összes tartalmát jelenítem meg. Ez különösen igaz egy sok oldalból álló weboldal esetén.

A dizájnért a CSS (Cascading Style Sheets) állományok feleltek, illetve segítségül hívtam a Bootstrap 4 könyvtárát is, melyet beintegráltam a fájl állományaim közé. Ez a könyvtár rendkívül sok segítséget nyújtott és viszonylag rövid idő alatt jól elsajátítottam használatát. Célul szolgált azért is, hogy reszponzívvá tegyem weboldalam, így kisebb képernyő méretekben, más eszközökön is (például mobil) gond nélkül használható lett.

A reszponzivitás fontos eleme egy weboldalnak, mivel egyre több felhasználó nem csak 1 eszközt használ manapság a weboldalak böngészéséhez és ezekhez az eszközök felbontásához igazodnia kell a weboldaloknak is, hogy modern és korszerűvé váljanak.

A következő kód sorral lehet közelebb vinni egy weboldalt a mobil-barát jelzőhöz a fejlécben felhasználva:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

1.sz. Kódrészlet: *Az adott oldal mobil baráttá tétele*

Az oldal elkészítéséhez felhasználtam az iskolában és iskolán kívül tanult imént említett PHP ismeretemet is. Különféle lekérdezéseket, műveleteket hajtottam végre a segítségével a weboldal és a MySQL adatbázisom között. Így tudtam kéréseket továbbítani a tárolt adatok felé és feldolgozni azt.

Ezek mellett JavaScript programozási nyelvet is használtam az iskolában tanultak szerint. Segítségével egy népszerű könyvtárat tudtam alkalmazni, ez a jQuery. Segítségével további műveleteket tudtam végrehajtani, úgynevezett AJAX (Asynchronous JavaScript and XML) kéréseket. Rendkívül előnyös tud lenni a használata. Fő pozitív tulajdonságának tartom, hogy oldal újra töltődés nélkül megtudjuk változtatni az oldal tartalmát. Így ezt kihasználva olyan funkciókat hoztam létre oldal frissülés nélkül, mint a kosárból való törlés, vagy a kosár ürítése. Ezek egy szempillantás alatt végre is hajódnak.

3.1 honlap felépítése

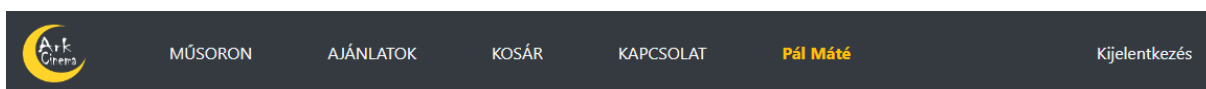
A weblapot igyekeztem egységes kinézettel elkészíteni. A sötétebb színek dominálnak, ami felidézheti a látogatóban a sötét mozi termek varázsát.

Először egy menüsávot hoztam létre, melybe a felhasználó elnavigálhat a különböző információk és funkciók között könnyedén. Miután elkészítettem egyszer, a többi oldalon elég volt már csak meghívnom az „include” parancs segítségével a következőképp:

```
include('config/functions.php');
```

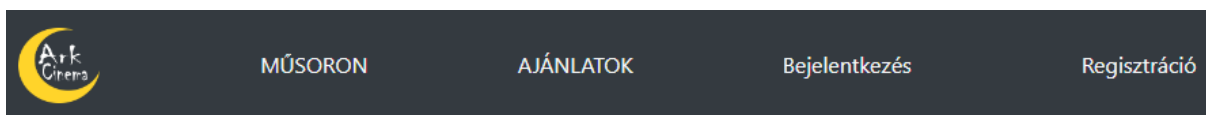
2.sz. Kódrészlet: Egy fájl tartalmának injektálása egy másikba

A „functions.php” nem csak szimplán megjelenítette a menüsávot, hanem eldöntöttem, hogy az oldalon van-e bejelentkezett felhasználó vagy nincs. Ennek függvényében változott a menüsáv által kínált lehetőségek sokasága. Ha volt bejelentkezett felhasználó a következő opciók álltak rendelkezésre:



2.sz.Ábra: Bejelentkezett menüsáv

Ha nem, akkor pedig:



2.1 sz. Ábra: Kijelentkezett menüsáv

3.1.1 Menüsáv elemei

A cég logójára kattintva a főoldalra kerülünk, ami egy szimpla kezdőoldalként funkcionál. Az időközönként váltakozó képek egy része linkként is szolgál néhány menüsávhoz.

A műsoron menüpontba megtalálható az éppen futó filmek listája, melyről információkat szerezhet a felhasználó.

Az ajánlatok fül alatt ki lehet választani, hogy melyik vetítésre és milyen időpontba szeretne a látogató jegyet foglalni. A kosárba rakás funkció csak akkor érhető el, ha rendelkezik egy regisztrált fiókkal és be van jelentkezve az illető.

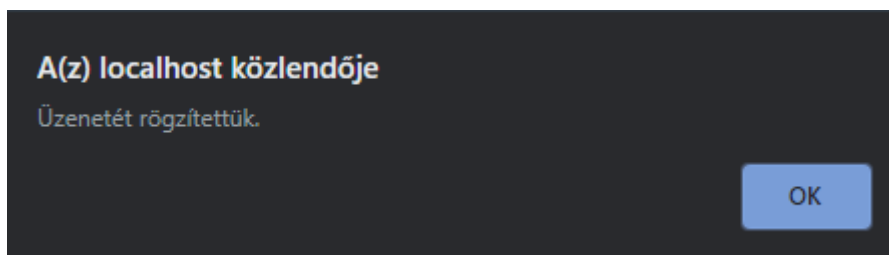
A regisztrációra kattintva megjelenik egy bemeneti mezőkből álló űrlap és miután azokat sikeresen kitöltöttük lehetőségünk lesz a bejelentkezésre. Itt meg kell adni egy kitalált

felhasználónevet, jelszót, meg kell adnunk a hivatalos nevünket, lakcímünket és egy e-mail címet.

Miután regisztráltunk sikeresen, a bejelentkezés opciót választva megadhatjuk felhasználó nevünk és jelszónkat. A hitelesítés után további menüpontok közül választhatunk.

A kosár tartalmazza mindazon jegyeket, melyek az ajánlatoknál bele lettek helyezve. Itt egy részletes, megrendelés előtti információs táblázat kerül a felhasználó elé, melynél letudja ellenőrizni, hogy a kívánt jegy pontosan azokkal a paraméterekkel van-e ellátva, amit szeretne. Ha nem, akkor lehetősége van törölni az összes jegyet, akár egyesével, akár mindet egyszerre. Ha mindent rendben talál a filmezni kívánó, akkor egyszerűen a megrendelés gombra kattintva leadhatja rendelését, ezzel hozzászúrva új adatokat az orders és items táblákhoz az adatbázisba, amit könnyen ellenőrizhet az arra jogosult.

Különös figyelmet szenteltem a jegyfoglalás rendszerének elkészítésére, hogy tükrözzem a valóságot egy ilyen folyamatkor. Például, ha kiválaszt a felhasználó egy hónapot, legyen az a február akkor az adott év februárjának megfelelő napszámát listázza ki és azok közül választhatunk dátumot. Az is fontos volt, hogy egyszerre 2 ugyanolyan paraméterekkel megadott jegyet ne lehessen a kosárba helyezni. Akkor sem, ha csak a jegytípusa változik, hisz az a vetítés adott széke attól függetlenül foglalt már ideiglenesen. A kapcsolatnál kapcsolatba lehet lépni a céggel, üzenetet írhatunk, mely a küldés gomb lenyomása után feltöltődik a contact adatbázis táblába és az asztali alkalmazáson keresztül az arra jogosultak megtekinthetik, válaszolhatnak a megadott e-mail címre. Erről a felhasználó a következő értesítést kapja:



3.sz. Ábra: *Visszajelzés a küldés gomb lenyomása után*

A Sárgán jelzett hivatalos névnél lehetőség van a felhasználói adatok módosítására, ha megváltozna az illető lakcíme, hivatalos neve, vagy csak egy új e-mail címet, felhasználónevet szeretne használni a továbbiakban. A lakcímnek a következő formai követelménynek kell megfelelnie mind a regisztrációkor, mind az adatok módosításakor: 4 számjegyű, 0-val nem kezdődhető irányítószám, szököz után a város neve, ahol él, utca és a házszám.

Példa: 1234 Város kitalált utca 10

Erre felhívtam a felhasználó figyelmét is a következőképp:

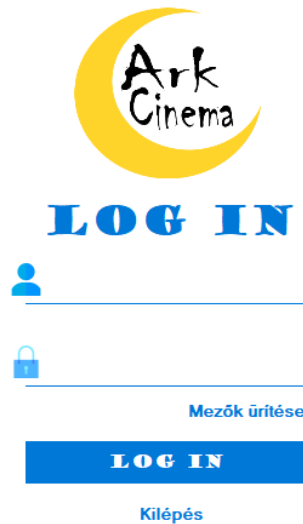


4.sz.Ábra: *Lakcím formai követelménye példával*

A kijelentkezéssel visszakerülünk a főoldalra és újra csak látogatók leszünk kevesebb lehetőséggel.

4. Az asztali alkalmazás

A program indulásakor egy bejelentkezési felület jelenik meg, ahol a felhasználó megadhatja a felhasználó nevét és jelszavát. Emellett lehetőség van üríteni a bemeneti mezőket, illetve a programból kilépni is.



5.sz.Ábra: Bejelentkezés ablak az asztali alkalmazásnál

Sikeres bejelentkezés után a program átirányítja a fő menübe a bejelentkezni vágyót.

A menüszalag segítségével a további funkciók betöltésére van lehetőség:

4.1 Menüsáv elemei

Fájl/Kilépés:

- Itt egyszerűen kiléphetünk a programból. Azért nem szimplán a menüsávban helyezkedik el ez a funkció kilépés névvel, hogy véletlen ne kattinthatson rá a felhasználó, mikor épp dolgozik.

Filmek/Műsoron:

- Kiválasztása után megjelenik egy új ablak az MdiContainer-ben, mely kilistázza az összes adatbázisban található filmet minden egyéb jellemzőjével együtt, ahogy az az adatbázis movies táblájában is található.
- A kilistázott elemek oldalakra vannak tördelve és lapozási lehetőség van. Ez azért előnyös, mert egyszerre kisebb erőforrást használ fel a számítógép, mivel kevesebb adatot kell feldolgoznia és a képernyőre rajzolnia a GUI (graphical user interface) felületen. Illetve további előnye, hogy átláthatóbb a táblázat, amit megjelenít.
- Lehetőség van szelektálni is egy kereső sáv és gomb segítségével film cím alapján. Erre fel is hívtam a figyelmet.

6.sz. Ábra: Keresősáv a műsoron lévő filmek menüpontban

- Ezek mellett tartalmaz az ablak egy számlálót is, ami kiírja, hogy mennyi elem került kilistázásra. Ha ezeket módosítjuk, akkor dinamikusan ez a szám is változni fog.
- Tudunk sorba rendezni is a borítót kivéve az összes tulajdonság alapján negatív és pozitív irányba is. Ezt úgy tehetjük meg, hogy rákattintunk a kívánt címkére a táblázat fejrészében. Alapértelmezetten a film cím alapján van sorba rendezve.
- Mivel ez egy adminisztrációs felület, ezért lehetőségünk van új filmek beszúrására, meglévők szerkesztésére, törlésére is. Az Új gombot megnyomni megjelenik egy új ablak, ahol beszúrhatunk egy új filmet minden paraméterét megadva. A sikeres kitöltés után kattintsunk az OK gombra. Ha mégsem szeretnénk ezt megtenni miután megnyitottuk, csak kattintsunk a Mégse gombra.

7.sz. Ábra: Új film beszúrása vagy meglévő szerkesztése ablak

- Ha szerkeszteni szeretnénk ugyanezzel az ablakkal kell dolgoznunk, csak előhívni úgy tudjuk, hogy kiválasztunk egy (akár több) sort, majd jobb egérgomb, szerkesztés.
- Törölni tudunk ugyanezzel a módszerrel, vagy a menüsávban található törlés gombbal miután kiválasztottunk egy vagy több sort, amit elszeretnénk távolítani.

Azonban ezek a CRUD műveletek csak a nézetben tesznek változtatásokat mindaddig, amíg nem nyomunk a Mentés gombra. Ez a gomb felel az adatbázissal való kommunikációért. Minden művelet, amit az asztali alkalmazásba végzünk el, az csak és csak akkor érvényes, ha ezt lenyomtuk. Ezzel elkerülhetők különféle véletlen vagy hibás

változtatások. Nagyobb eséllyel veszi észre a felhasználó a hibát, amit vétett a hiba elkövetése után, de ha nem mentette még el, akkor nincs miért aggódnia.

Felhasználók/Összes felhasználó:

- Tartalmazza az adatbázisban található user tábla mezőit és annak értékeit, kivéve a jelszót a biztonság érdekében.
- Elvégezhető keresés az adatok között hivatalos név, felhasználónév, lakcím alapján. Továbbá tartalmazza a törlést, szerkesztést is. A sorba rendezhetőség, oldalakra tördelés és az adatsor száma is szintén elérhető és alkalmazhatjuk őket. Minden módosítás után kérem mentsen a változtatások megőrzése érdekében!

Rendelések/Tételek:

- Tartalma az adatbázis items táblája.
- A keresés, sorba rendezés, oldalra tördelés, összes adatsor és a törlés, szerkesztés szintén elérhető művelet. Minden módosítás után kérem mentsen a változtatások megőrzése érdekében!

Rendelések/Foglalások:

- A keresés, sorba rendezés, oldalra tördelés, összes adatsor és a törlés szintén elérhető művelet. Minden módosítás után kérem mentsen a változtatások megőrzése érdekében!

Kapcsolat/Felhasználói üzenetek:

- A keresés, sorba rendezés, oldalra tördelés, összes adatsor és a törlés szintén elérhető művelet. Minden módosítás után kérem mentsen a változtatások megőrzése érdekében!

4.2 Objektum orientált programozás (OOP)

Igyekeztem az asztali alkalmazásom objektum orientáltan elkészíteni. Ennek néhány előnye az átláthatóság, a könnyen szerkeszthetőség.

Az objektum orientált programozás egy programozási paradigma, mely objektumokból építi fel a programot. A program ezután az egyes objektumok kommunikációjával válik működőképpé.

Legfontosabb alapelvei:

Konstruktorok:

- Ez egy speciális metódus, amit objektumok esetén alkalmazunk. Konkrétan az objektumok létrehozásakor fut le.

- Feladata kezdő érték adás az osztályon belüli változóknak.
- Minden osztály legalább egy konstruktor metódust tartalmaz. Ha ezt nem definiáljuk, akkor a fordító generál egy paraméter nélkülit.

```
namespace CinArk.Models
{
    20 references
    partial class movies
    {
        1 reference
        public movies(string title, string genre, string rating, string lenght, string release_date, int room, byte[] img)
        {
            this.title = title;
            this.genre = genre;
            this.rating = rating;
            this.lenght = lenght;
            this.release_date = release_date;
            this.room = room;
            this.img = img;
        }

        0 references
        public movies()
        {
            this.title = title;
            this.genre = genre;
            this.rating = rating;
            this.lenght = lenght;
            this.release_date = release_date;
            this.room = room;
            this.img = img;
        }
    }
}
```

3.sz. Kódrészlet: Paraméteres (első) és paraméter nélküli (második) konstruktor

Legfontosabb jellemzői:

- Nincs visszatérési értékük.
- Nevük megegyezik az osztály nevével.
- Fogadhatnak paramétert.
- Több is lehet belőlük, viszont jól meg kell őket különböztetni.

Osztály alapú szerkezet és különböző típusú metódusok használata:

- Osztály alapú nyelvekben az osztályokat előre kell definiálni, az objektumokat ezekből példányosítással kapjuk.

```
private ArkContext db = new ArkContext();
```

3.1.sz. Kódrészlet: Az ArkContext példányosítása

dinamikus kötések és üzenetátadások:

- Nem a kliens, hanem az objektum feladata megválasztani, hogyan reagáljon egy metódushívásra. Ezt tipikusan futás időben végzi el, és a metódushívást a hozzá társított táblából választja ki.
- A metódushívást tekintik üzenetátadásnak is, ahol a kliens a kötésben részt vevő objektumnak küld üzenetet.

egységbe zárás:

- Az egységbe zárás azt fejezi ki, hogy az összetartozó adatok és függvények, eljárások együtt vannak, egy egységbe tartoznak.

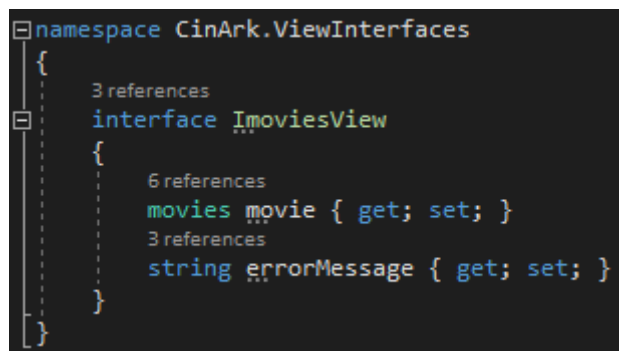
- További fontos fogalom az adatelrejtés, ami azt jelenti, hogy kívülről csak az férhető hozzá közvetlenül, amit az objektum osztálya megenged. Ez fontos ahhoz, hogy megelőzze a nem kívánt kapcsolatok kialakulását, megkönnyítse a kód értelmezését, és elkerülje az adatok ellenőrizetlen terjedését.

kompozíció, öröklődés:

- Az objektumok lehetnek más objektumok mezői, ez az objektumok kompozíciója.
- Majdnem minden osztály alapú nyelv támogatja az öröklődést. A szülő osztály minden adata és metódusa jelen van a gyermek osztályban (alosztálynak is nevezik) is.

interfészek használata:

- Az interfészek tulajdonképpen absztrakt osztályok.
- Nem tartalmazhatnak megvalósítási részleteket, csak előírhatják bizonyos metódusok jelenlétét, illetve konstansokat definiálhatnak.



```

namespace CinArk.ViewInterfaces
{
    3 references
    interface IMoviesView
    {
        6 references
        movies movie { get; set; }
        3 references
        string errorMessage { get; set; }
    }
}

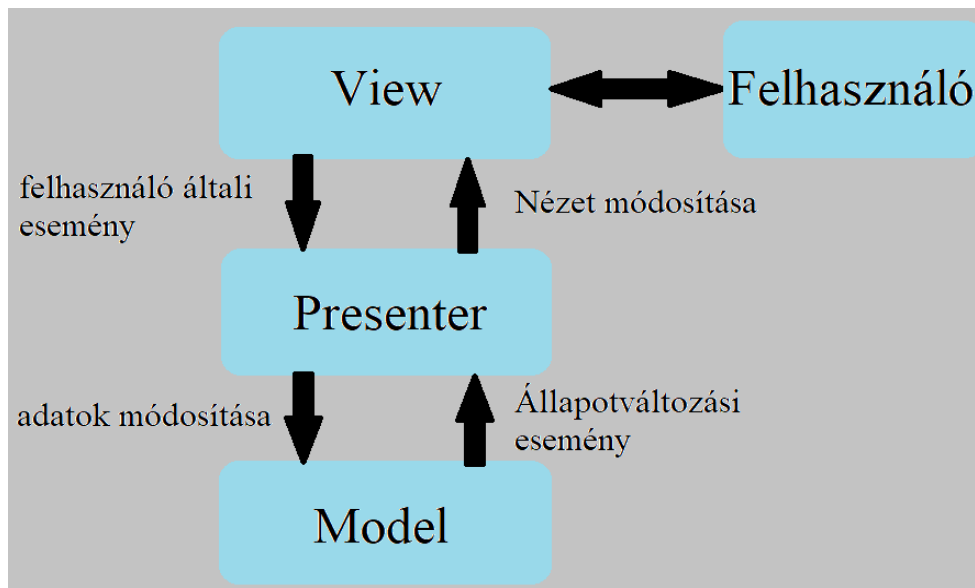
```

4.sz. Kódrészlet: *Interface példa*

4.3 Modell-nézet-prezenter (MVP)

Az MVP egy szoftverfejlesztési minta, amely a Modell-nézet-vezérlő (MVC) mintából alakult ki. Az MVP-modell főleg olyan alkalmazásokban nyújt jelentős előnyöket, ahol komplex adathalmazokon kell műveleteket végezni, és ezek eredményeit a felhasználó elé tárni.

Ezzel az ábrával szeretném szemléltetni a különböző részek feladatát:



8.sz. Ábra: MVP minta

- 1) A **Model** felel az adatok tárolásáért.
- 2) A **View** jeleníti meg az adatokat a felhasználónak, illetve a képernyőn történt bemeneti értékeket itt fogadja a program.
- 3) A **presenter** egyfajta vezérlő, ami összegyűjti, tovább irányítja az adatokat, kéréseket, metódusokat és még akár sok mást a programban, hogy minden eljusson a megfelelő helyre.

Ezen felül létrehoztam egy **repository** réteget, mely a Model-ből érkező adatokkal végez műveleteket. Ilyen volt a keresés, sorba rendezés, oldalakra tördelés funkció is például. A különböző adatbázis lekérdezések is itt kaptak helyet, melyet könnyedén véghez tudtam vinni a LINQ segítségével.

4.4 LINQ (Language-Integrated Query)

A leggyakrabban használt lekérdezési operátorok a programban:

- Where

Itt megadhattam különböző feltételeket, melyek hatására egyfajta szűrés valósult meg és a kívánt adathalmazomat kaptam vissza.

- OrderBy

Növekvő sorrendbe rendeztem az adataim a megadott paraméter alapján.

- OrderByDescending

Csökkenő sorrendbe rendeztem az adataim a megadott paraméter alapján.

Példa egy LINQ lekérdezésre a programból:

```
query = ascending ? query.OrderBy(x => x.title) : query.OrderByDescending(x => x.title);
```

5.sz. Kódrészlet: *Növekvő vagy Csökkenő sorrendbe rendezés title (cím) alapján*

Ahhoz, hogy létrehozzak egy lekérdezést átkellett alakítanom egy változó típusát lekérdezéssé.

Ehhez a AsQueryable kódot kellett alkalmaznom a következő képpen:

```
var query = db.movies.OrderBy(x => x.title).AsQueryable();
```

6.sz. Kódrészlet: *query változó lekérdezéssé alakítása + növekvő sorrendbe rendezés cím alapján*

4.5 Státusz

A főmenü alsó részén található egy „statusStrip”, mely megjeleníti a bejelentkezett felhasználó nevét.

Bejelentkezve: admin

9.sz. Ábra: *A bejelentkezett felhasználó nevének megjelenítése*

5. Tesztelés

A program megírása során folyamatosan teszteltem a folyamatok működését, milyen paraméterek, értékek továbbítódnak az adott metódusoknál. A szoftvertesztelés célja a hibák feltárása. Minél hamarabb találjuk meg a hibákat, annál könnyebb kijavítani azokat, mellesleg rengetek időt spórolhatunk meg.

Mind az asztali alkalmazást, mind a weblapot folyamatos tesztelésnek vetettem alá. Az én szakdolgozatomban a fő rész az a weblap volt, így ott különös hangsúlyt fektettem ezekre. Fontos volt olyan modulokat/validációkat beépítenem például az űrlapokba, hogy ne lehessen üresen hagyott mezővel vagy hibás formátumban megadott kéréseket indítani. Ennek elkészítése sok tesztelést vont maga után. Néhány űrlapnál azért véltem fontosnak a formai követelményeket, hogy egy egységes, könnyen átlátható monitorozási lehetőséget nyújtson az ezt ellenőrzők számára.

A weblap kinézetének tesztelése is sok időt vett el, mert az elkészítés során tanultam meg például a Bootstrap 4 könyvtárát megfelelően használni és eleinte nehézséget okozott, tesztelni kellett.

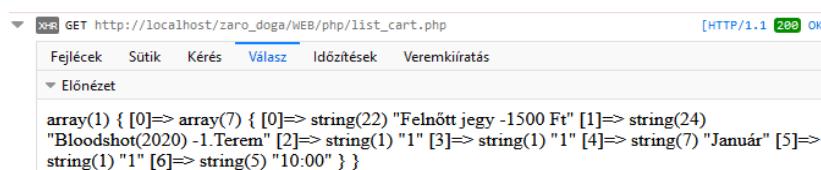
A PHP-ban és JavaScript-ben megírt kódok még ennél is több elemzésnek voltak kitéve, a „var_dump” és a „console.log” parancsok végig a segítségemre voltak és információkkal láttak el a különböző változók, tömbök értékeiről. Ezen paraméterek egy, a böngésző által előhozható konzolon jelent meg, ahol sok más, a weboldallal kapcsolat információ is megtekinthető.

Én a Mozilla Firefox böngészőt használtam erre a célra, ahol az F12 billentyűgomb lenyomásával megjelentek a kívánt információk. Azért ezt választottam, mert számomra könnyen átlátható és gyorsan elsajátítható volt a használata.

```
$cart = $_SESSION['cart'];  
  
var_dump($cart);
```

7.sz. Kódrészlet: *var_dump* használata a *\$cart* változóval

Ez a böngésző konzolján a következő információkkal szolgál:



10.sz.Ábra: *Firefox böngésző konzoljában a list_cart.php fájlban lévő var_dump(\$cart) tartalma*

6. Fejlesztési lehetőségek

A záródolgozat során rangsorolnom kellett prioritás szerint mindent, hogy mi az ami elengedhetetlen a weblaphoz vagy az asztali alkalmazáshoz egyaránt. Több dolog is volt amikre sajnos nem maradt időm, illetve össze gyűjtöttem olyan további lehetőségeket is, amik a fejlesztés során, után eszembe jutottak és véleményem szerint érdemes lenne beépíteni, ezzel tovább növelve az összminőséget.

6.1 A weblap további fejlesztési lehetőségei

Összességében erre a modulra helyeztem a nagyobb hangsúlyt, mind funkcionalitás és kinézet szerint is.

1. Menet közbe rájöttem, hogy a jegyfoglalásnál egy grafikus felületű rendszer jobb megoldás lett volna, mint a jelenlegi legördülő listát táblázat. Jobban átlátható és egy mozi weboldalán sokat számít a megjelenés.
2. Szerettem volna lehetőséget biztosítani a jelszó cseréjére is egy hasonló űrlappal, mint a profil adatok módosítása, ám erre nem maradt időm.
3. A műsoron lévő filmeknél be lehetne építeni a jövőben egy keresőt, amivel könnyedén elnavigálhatja magát a látogató a megnövekedett filmek választékában. Szintén ebben a menüpontban érdemes lenne minden filmnél beiktatni egy Részletek gombot, ami elvisz minket egy új oldalra, ami egy részletesebb leírást, illetve egy beágyazott bemutató videót is nyújt a filmekről, ezt eleinte még nem gondoltam olyan fontosnak, de a valóságban ez is sokat jelent, mert segíti a felhasználót, hogy melyik műsort válassza.
4. Szerettem volna egy teljesen új menüpontot, ami egy egyedi megoldást nyújtana a többi hazai mozival szemben. Lehetőséget nyújt a jegyfoglalás mellett a helyi büfé szolgáltatásait is igénybe venni. Ezt úgy képzeltem el, hogy a felhasználónak lehetősége van kiválasztani azokat az ételeket, italokat előre, amiket szívesen fogyasztana az előadás közben. A foglalás árához ezeknek a tételeknek az ára hozzáadódna. Ez a helyszínen úgy történne, hogy lenne egy VIP sor, ahol zökkenőmentesen, a hosszú sorba állást elkerülve kézhez kaphatnánk az előre megváltott ételeink, italaink.
5. A jegy, étel-ital foglalás nem minden. Még gyorsabbá tehetnénk a folyamatot, ha nem csak foglalni lehetne ezeket a cikkeket, hanem megvásárolni online. Ehhez szívesen beépítettem volna egy bankkártyás fizető rendszert, ám idő hiányában

ezeket az ismereteket még nem tudtam elsajátítani, így elmaradt, de a jövőben mindenféleképp érdemes funkció lenne.

6. A jövőben érdemes lehet beépíteni egy látogatói értékelés, komment szekciót, ahol azok a vásárlók, akik az adott filmre jegyet vásároltak 1 hozzászólást, illetve 1 értékelést (például 1-5 skálán) tudnak hozzáfűzni. Ennek vannak hátulütői is, de ha magas a színvonal, akkor csak pozitív irányba mozdíthatja el az emberek véleményét a cégről és a hely által bemutatott filmekről.
7. Tervbe volt ajándékjegy foglalás, vásárlási lehetőség is, amihez nem lett volna szüksége másra a regisztrált felhasználónak, mint az illető e-mail címére és hivatalos nevére. Az e-mail címre érkezett volna meg valamilyen formában a jegyhitelesítő, amit helyben felmutatva felhasználhatja jegyét az illető. Maga a jegyvásárlás és jegyfoglalást is így képzeltem el megvalósítani, de egyelőre csak a hivatalos névvel egyező személyi igazolványon szereplő névvel hitelesíthető a jegyfoglalás a kasszáknál.
8. Az oldal reszponzív tulajdonsága sem 100% még, így azon is a későbbiekben érdemes finomítani.
9. Volt olyan ötletem is, hogy hírlevél biztosítása a regisztrált felhasználóknak, ha igénylik, melyben aktuális információkat kaphatnak a legújabb filmekről, az esetleges adatvédelmi és a weboldal működéséről kapcsolatos változásokról.
10. Egy szintén fontos dolog a megrendelt jegy validáció, melyet szerettem volna beépíteni az alkalmazásomba.
11. Utolsó, egyben nagyon fontos változtatási ötletem az az, hogy a PHP kódjaim és aminek még szükséges, érdemes lenne felvértetni objektum orientált programozási paradigmával. Emellett egy népszerű szoftvertervezési mintával is, például az MVC-vel. Ezzel sokkal átláthatóbb kódot eredményeznék nem csak saját magam, de más fejlesztők számára is. Nemzetközileg elfogadottabb eredményt kapnék és ha később módosításokra van szükség, akkor nem én leszek az egyetlen, aki érti, amit lát néhány esetben az átláthatatlanság végett.

6.2 Az asztali alkalmazás további fejlesztési lehetőségei

Összegyűjtöttem néhány, szerintem fontos fejlesztési ötletet, melyek elkészítése a jövőben minőségibbé tehetné az alkalmazást.

1. Telepítő készítése a komplett alkalmazáshoz.
2. Adatok megjelenítésének lehetősége adatbázis elérés nélkül is, például egy .xml vagy .csv fájlból beolvasva.
3. A felhasználói üzenetek menüpontnál közvetlen lehetőség e-mail küldésére az üzenet küldőknek. Minden sorban lenne egy Válasz gomb, melyre rákattintva egy ablak jelenne meg, ahol válaszolhatunk a felhasználóknak.
4. Minden jegyfoglalás, ami fél órával az előadás előtt nem kerül hitelesítésre, az törlődik és újra szabad hely lesz. Ehhez segítségül létrehoznék egy „hitelesítve” névvel ellátott CheckBox-ot, amit, ha nem pipál be az adminisztrátor az előadás kezdete előtt 30 perccel, akkor az adott tétel törlődni fog és újra szabad lesz ezáltal a hely.
5. Továbbá az előadás kezdete után eltelt 3 óra után az összes jegyfoglalás átkerül egy archivált fájlba, táblázatba, ami egy bizonyos ideig eltárol mindent. Ez biztonsági szempontból fontos. Például, ha valami károkozás történne egy teremben, akkor lehessen szűkíteni ezen tételek alapján, hogy mely felhasználók tartózkodtak a teremben akkor.

7. Használati útmutató

Ebben a pontban egy utcáról betévedő embernek is lehetősége van elsajátítani a cinemaArk asztali alkalmazásának működését és könnyen használhatóvá teheti magának.

7.1 Bevezetés

Üdvözljük a cinemaArk felhasználók között!

A következő sorokban leírjuk önnek, hogy milyen lépéseket kell végrehajtania ahhoz, hogy használhassa asztali alkalmazásunkat és részletes útmutatást kapjon minden funkcióról.

Jó munkát!

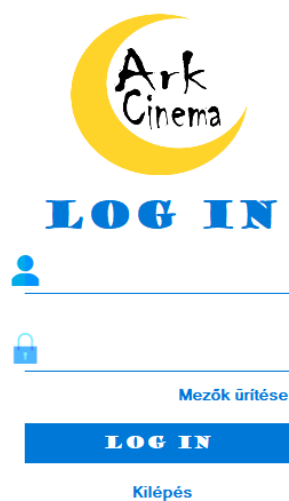
7.2 Telepítés

Minimális rendszerkövetelmények:

- Processzor: 1 GHz
- RAM: 256 MB
- Operációs rendszer: Windows 10
- Microsoft .NET Framework 4.7.2
- Állandó internetkapcsolat

7.3 Bejelentkezés

A program indulásakor egy bejelentkezési ablak fogadja.

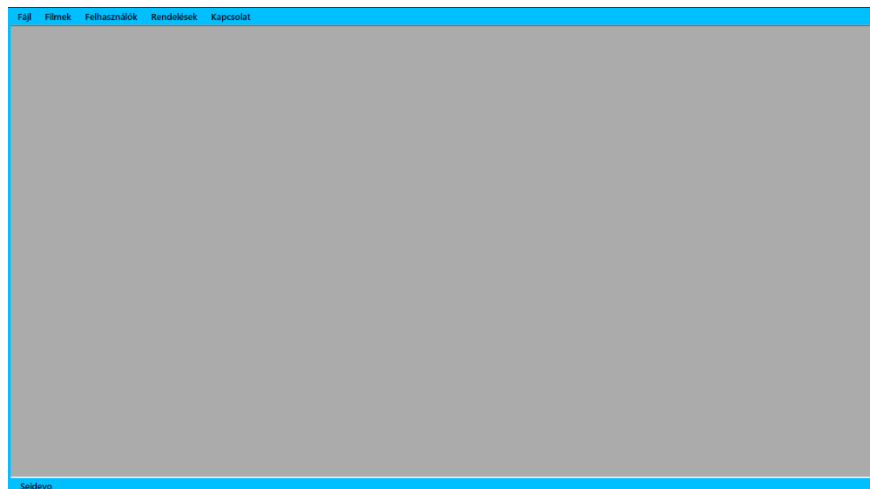


5.sz. Ábra: Bejelentkezés ablak az asztali alkalmazásnál

Itt kérjük adja meg felhasználónevét és jelszavát, amit a cég biztosított önnek.

7.4 Kezdőlap

A sikeres bejelentkezés a fő ablak jelenik meg, a következő formában:



11.sz. Ábra: *Az asztali alkalmazás fő ablaka*

A szürke részre fognak kerülni a különféle kilistázások, melyeket majd megtekinthet, módosíthat.

7.5 Menüsor használata

Fájl/Kilépés:

- Kilépés a programból.

Filmek/Műsoron:

- Az összes éppen futó film kilistázva megjelenik. Újat adhat hozzá, szerkesztheti a meglévőket és törölhet.
- Új adat **létrehozása**: Kattintson az új gombra, majd töltse ki a megjelenő ablakot és válassza az OK gombot. Végül a mentéssel véglegesítse.
- Meglévő adat **szerkesztése**: Válasszon ki egy (vagy több) sort és nyomja meg a jobb Klikket. Válassza a szerkesztés opciót, módosítsa a hibás adatokat, majd válassza az OK gombot. Végül a mentés gombbal véglegesítse.

Cím

Műfaj

Értékelés

Hossz

Megjelenés

Terem 1

Borító

Tallózás.. Mégse OK

7.sz. Ábra: Új film beszúrása vagy meglévő szerkesztése ablak

- **Törlés:** Válasszon ki egy sort és kattintson a törlés gombra, majd mentés vagy válasszon ki egy (vagy több sort) és kattintson rá jobb a jobb egérgombbal, majd törlés. Végül a mentés gombbal véglegesítse.
- A **kereső sáv** segítségével szűrheti a találatokat, illetve a táblázat fejrésében lévő címkékre kattintva sorba is rendezheti azokat az adott címke szerint.

Minden művelet után kérjük kattintson a mentés gombra a változások megtartása érdekében!

Felhasználók/Összes felhasználó:

- Az összes felhasználó kilistázva megjelenik.
- Meglévő adat **szerkesztése:** Válasszon ki egy (vagy több) sort és nyomja meg a jobb Klikket. Válassza a szerkesztés opciót, módosítsa a hibás adatokat, majd válassza az OK gombot. Végül a mentés gombbal véglegesítse.
- A **kereső sáv** segítségével szűrheti a találatokat, illetve a táblázat fejrésében lévő címkékre kattintva sorba is rendezheti azokat az adott címke szerint.
- **Törlés:** Válasszon ki egy sort és kattintson a törlés gombra, majd mentés vagy válasszon ki egy (vagy több sort) és kattintson rá jobb a jobb egérgombbal, majd törlés. Végül a mentés gombbal véglegesítse.

Minden művelet után kérjük kattintson a mentés gombra a változások megtartása érdekében!

Rendelések/Tételek:

- Az összes tétel kilistázva megjelenik.
- Meglévő adat **szerkesztése**: Válasszon ki egy (vagy több) sort és nyomja meg a jobb Klikket. Válassza a szerkesztés opciót, módosítsa a hibás adatokat, majd válassza az OK gombot. Végül a mentés gombbal véglegesítse.
- A **kereső sáv** segítségével szűrheti a találatokat, illetve a táblázat fejrészában lévő címkékre kattintva sorba is rendezheti azokat az adott címke szerint.
- **Törlés**: Válasszon ki egy sort és kattintson a törlés gombra, majd mentés vagy válasszon ki egy (vagy több sort) és kattintson rá jobb a jobb egérgommbal, majd törlés. Végül a mentés gombbal véglegesítse.

Minden művelet után kérjük kattintson a mentés gombra a változások megtartása érdekében!

Rendelések/foglalások:

- Az összes jegyfoglalás kilistázva megjelenik.
- Meglévő adat **szerkesztése**: Válasszon ki egy (vagy több) sort és nyomja meg a jobb Klikket. Válassza a szerkesztés opciót, módosítsa a hibás adatokat, majd válassza az OK gombot. Végül a mentés gombbal véglegesítse.
- A **kereső sáv** segítségével szűrheti a találatokat, illetve a táblázat fejrészában lévő címkékre kattintva sorba is rendezheti azokat az adott címke szerint.
- **Törlés**: Válasszon ki egy sort és kattintson a törlés gombra, majd mentés vagy válasszon ki egy (vagy több sort) és kattintson rá jobb a jobb egérgommbal, majd törlés. Végül a mentés gombbal véglegesítse.

Minden művelet után kérjük kattintson a mentés gombra a változások megtartása érdekében!

Kapcsolat/Felhasználói üzenetek:

- Az összes üzenet, mely a felhasználóktól érkezett megjelenik.
- A kereső sáv segítségével szűrheti a találatokat, illetve a táblázat fejrészában lévő címkékre kattintva sorba is rendezheti azokat az adott címke szerint.
- **Törlés**: Válasszon ki egy sort és kattintson a törlés gombra, majd mentés vagy válasszon ki egy (vagy több sort) és kattintson rá jobb a jobb egérgommbal, majd törlés. Végül a mentés gombbal véglegesítse.

8. Összegzés

Számomra a szakdolgozat fejlesztése, megírása egy nagyon jó fejlődési és a szakma felé irányuló morál növelő lehetőség volt. Többször is elakadtam, de a nehézségeket hátrahagyva, nem hagytam annyiban a problémákat és ezáltal a hozzáállásom is jobb lett. Kitudtam élni a kreativitásom is, ami miatt szívesen álltam neki a nehezebb részeknek is. A projekt elkészítése során több weboldal is a segítségemre volt, ezeket részletezem a forrásokban.

Kicsit sikerült belekóstolnom milyen lehet egy szabadúszó fejlesztőnek a munkája, ami még ha nem is a tényleges egész képet adja vissza, első ránézésre szívesen foglalkoznék ebben a formában a programozással a jövőben, mint egyéni vállalkozó. Persze még rengeteget kell tanulnom, de úgy gondolom a befektetett energia és idő meg fog térülni. Úgy érzem még nem érte el a „release” verziót a weboldal és az asztali alkalmazás sem. Túl sok minden van, amit szívesen beleépítenék, de ez nagyon sok idő. A későbbiekben erre lehet sort is kerítek egy új fajta, objektum orientált szemlélettel.

9. Felhasznált irodalom

Wikipédia:

- OOP ismeretek -> https://hu.wikipedia.org/wiki/Objektumorientált_programozás

csharptutorial:

- OOP ismeretek -> <https://csharptutorial.hu/docs/hellovilag-hellosharp/5-osztalyok-objektumok/metodusok-konstruktor-peldanyositas/>

Dr. Nyári Tibor:

- Adatbázis ismeretek- órai jegyzet
- <https://summers.hu/szakkep.html>

Boros Bence:

- Programozási nyelvek - C# (2020) – tananyagok, órai feladatok

Bálint Róbert:

- Webfejlesztés – órai feladatok

9.1 Források

A szakdolgozatban felhasznált összes kép, mely nem a saját alkotásom:

készítő: Felix Mooneeram

- [cinema_section.jpg](#)

képet tartalmazó weboldal: rocketcdn.me

- [seats.jpg](#)

képet tartalmazó weboldal: influx

- [start.jpg](#)

képet tartalmazó weboldal: Innovia

- [snack.png](#)

Készítő: Freepik:

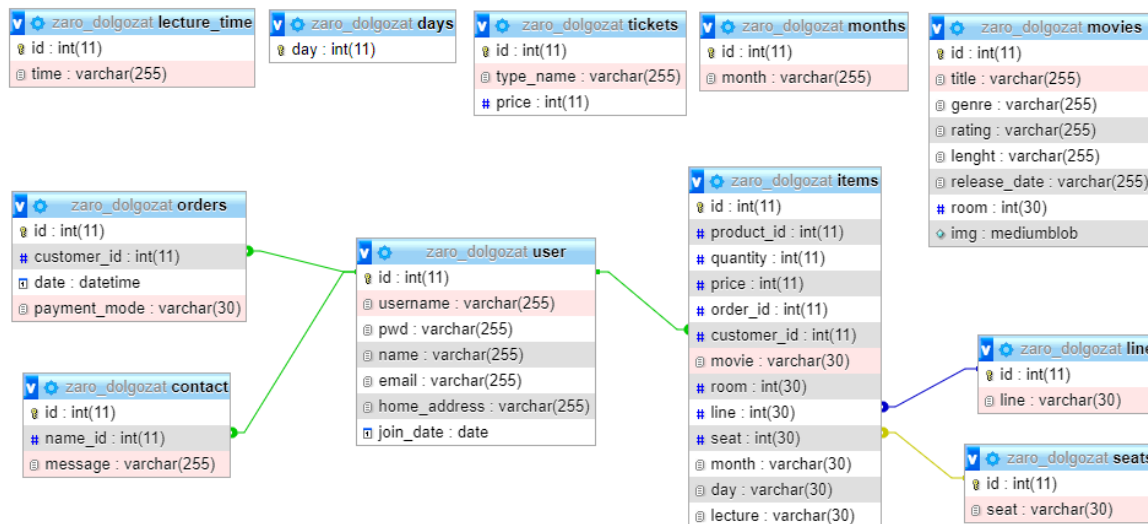
- [user.png](#)
- [lock.png](#)

Az összes film borító, ami szerepel a műsoron lévő filmek között, [ITT](#) került letöltésre.

Az összes URL link rövidítve lett az esztétika végett [ITT](#).

10. Mellékletek

A szakdolgozathoz tartozó, megértését elősegítő ábrák, diagramok.



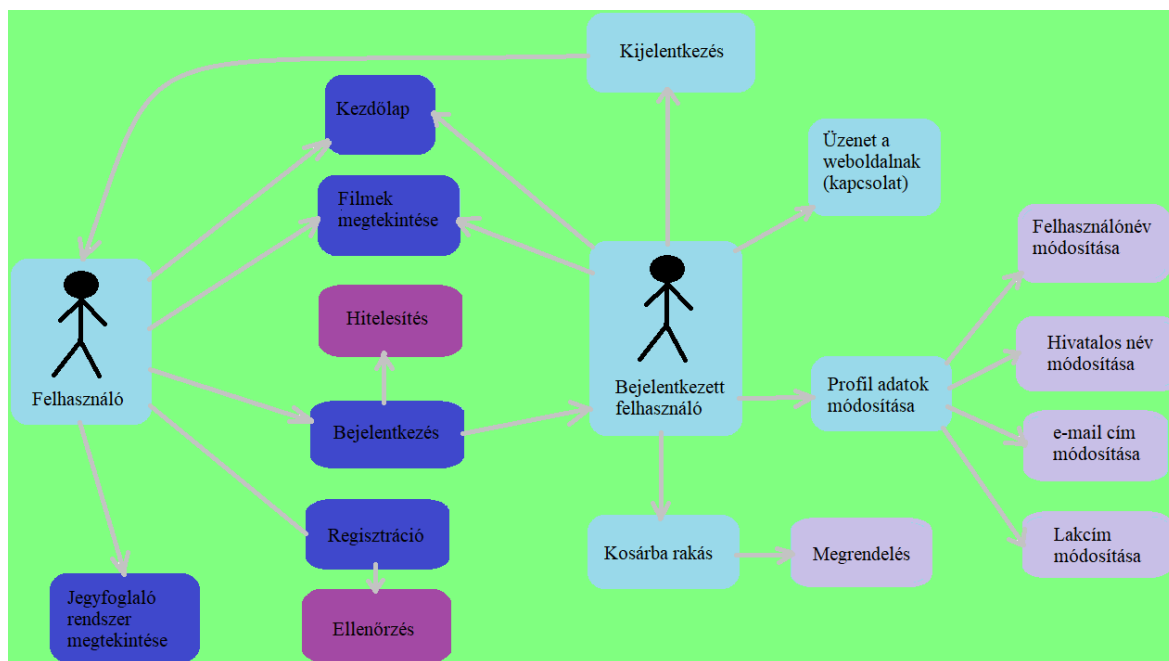
1.sz. Melléklet: Az adatbázisban használt táblák és kapcsolataik

ELŐADÁS JEGYEK	
Jegytípus	Felnőtt jegy -1500 Ft
Előadás	Bloodshot(2020) -1.Terem
Sor	1
Szék	1
Vetítési Hónap	Január
Vetítési nap	1
Időpont	10:00
Kosárba	

2.sz. Melléklet: A jegyfoglalás legördülő listás táblázata



3.sz. Melléklet: *A cinemaArk saját készítésű logója*



4.sz. Melléklet: *use-case diagram a weblapról*