

**SAPIENTIA ERDÉLYI TUDOMÁNY EGYETEM  
MAROSVÁSÁRHELYI KAR,  
INFORMATIKA SZAK**



**SAPIENTIA  
ERDÉLYI MAGYAR  
TUDOMÁNYEGYETEM**

eTickets .NET applikáció fejlesztése és dockerizálása

**DIPLOMADOLGOZAT**

Témavezető:  
Boróka Oltean-Péter,  
Tanársegéd  
dr. Farkas Csaba,  
Egyetemi tanár

Végzős hallgató:  
Vrencian Lilla

**2023**

**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA  
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,  
SPECIALIZAREA INFORMATICĂ**



**UNIVERSITATEA  
SAPIENTIA**

Dockerizarea si dezvoltarea applicatiei eTickets in .NET

**LUCRARE DE DIPLOMĂ**

Coordonator științific:  
Boróka Oltean-Péter,  
Asistent universitar  
dr. Farkas Csaba,  
Profesor universitar

Absolvent:  
Vrencian Lilla

**2023**

**SAPIENTIA HUNGARIAN UNIVERSITY OF  
TRANSYLVANIA**  
**FACULTY OF TECHNICAL AND HUMAN SCIENCES**  
**COMPUTER SCIENCE SPECIALIZATION**



**SAPIENTIA**  
HUNGARIAN UNIVERSITY  
OF TRANSYLVANIA

Dockerization and development of eTickets application in .NET

**BACHELOR THESIS**

Scientific advisor:  
Boróka Oltean-Péter,  
Teaching assistant  
dr. Farkas Csaba,  
Full professor

Student:  
Vrencian Lilla

**2023**

### **Declarație**

Subsemnatul/a Vrancian Lilla, absolvent(ă) al/a specializării  
Informatică, promoția 2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, Târgu Mureș  
Data: 05.06.2023

Absolvent

Semnătura.....

UNIVERSITATEA „SAPIENTIA” din CLUJ-NAPOCA  
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș  
Programul de studii: Informatică

Viza facultății:

## LUCRARE DE DIPLOMĂ

Coordonator științific: <b>Dr. Farkas Csaba</b> Îndrumător: <b>Oltean-Péter Boróka</b>	Candidat: Vrencian Lilla Anul absolvirii: 2023
---	---

**a) Tema lucrării de licență:**

Dockerizarea si dezvoltarea aplicației eTickets în .NET

**b) Problemele principale tratate:**

În lucrare este prezentată o aplicație numită eTickets, scrisă în cadrul framework-ului .Net și folosește o bază de date SQL. Scopul principal al lucrării este Dockerizarea software-ului. Scopul aplicației este dezvoltarea unei aplicații care facilitează vânzarea de bilete online.

**c) Desene obligatorii:**

Grafice despre structura bazei de date

Grafice despre structura codului

Desene despre design-ul aplicației

Diagramă de secvență

**d) Softuri obligatorii:**

Un software în .NET care implementează funcționalitățile asociate cu achiziționarea de bilete online, cum ar fi autentificarea, cumpărarea de bilete și vânzarea de bilete.

Software-ul include o bază de date SQL în partea din spate (backend) și aplicația este dockerizată.

**e) Bibliografia recomandată:**

.NET Framework, I. (2017). .NET Framework.

Tendulkar, S. S. (2022). Automated Dockerization of Web Applications Developed in Various Languages (Doctoral dissertation, Dublin, National College of Ireland).

**f) Termene obligatorii de consultații:** săptămânal

**g) Locul și durata practicii:** Universitatea „Sapientia” din Cluj-Napoca,  
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 416

Primit tema la data de: 20.06.2022

Termen de predare: 07.07.2023

Semnătura Director Departament

Semnătura responsabilului  
programului de studiu

Semnătura coordonatorului

Semnătura candidatului

# Kivonat

Dolgozatom témája az "eTickets" webapplikáció fejlesztése és dockerizálása .NET programozási nyelvben, ami lehetővé teszi a felhasználók számára, hogy jegyeket vásároljanak online különböző mozi termek filmjeire. Az online vásárlás népszerűsége az elmúlt években folyamatosan növekszik, és sok cégben bevett gyakorlattá vált. Ennek több oka is van, hogy miért is vezették be:

**Kényelem:** Az online vásárlás lehetővé teszi, hogy a vásárlók otthonról, munkahelyükről vagy bárbonnan hozzáférjenek az internethez, és könnyedén vásárolhassanak a kívánt termékekből vagy szolgáltatásokból. Nincs szükségük a hagyományos üzletekben való személyes jelenlétre vagy időpont egyeztetésére.

**Széles választék:** Az online térben a vásárlók hozzáférhetnek a világ számos tájáról származó termékek és szolgáltatások hatalmas választékához. Nem korlátozódnak a helyi üzletek árukészletére, és könnyen összehasonlíthatják az árakat, a márkákat és a minőséget a kényelmükön.

**Időmegtakarítás:** Az online vásárlás gyors és hatékony. A vásárlóknak nem kell végigjárniuk több üzletet vagy sorban állniuk a fizetéshez. Egyszerűen néhány kattintással vásárolhatnak és kifizethetnek egy terméket vagy szolgáltatást.

Ezen előnyök miatt sok cég dönt úgy, hogy az online vásárlást bevezeti vagy kiterjeszti, hogy elérje és kiszolgálja az internező vásárlók széles körét.

**Kulcsszavak:** *online, vásárlás, kényelem.*

# Rezumat

Tema tezei mele este dezvoltarea și dockerizarea aplicației web „eTickets” în limbaj de programare .NET, care permite utilizatorilor să cumpere online bilete pentru filme din diferite cinematografe. Cumpărăturile online au crescut în popularitate în ultimii ani și au devenit o practică obișnuită în multe companii. Există mai multe motive pentru care a fost introdus:

**Confort:** Cumpărăturile online le permit clienților să acceseze internetul de acasă, de la serviciu sau de oriunde și să cumpere cu ușurință produsele sau serviciile pe care le doresc. Nu trebuie să fie în persoană la magazinele tradiționale sau să facă o programare.

**Selectie largă:** în spațiul online, clienții au acces la o selecție uriașă de produse și servicii din multe părți ale lumii. Nu se limitează la inventarul magazinelor locale și pot compara cu ușurință prețurile, mărcile și calitatea din confortul propriei case.

**Economie de timp:** Cumpărăturile online sunt rapide și eficiente. Clienții nu trebuie să treacă prin mai multe magazine sau să stea la coadă pentru a plăti. Ei pot cumpăra și plăti pur și simplu pentru un produs sau serviciu cu câteva clicuri.

Datorită acestor avantaje, multe companii decid să introducă sau să extindă cumpărăturile online pentru a ajunge și a servi o gamă largă de cumpărători online.

**Cuvinte cheie:** *online, cumpărături, comoditate.*

# Abstract

The topic of my thesis is the development and dockerization of the "eTickets" web application in .NET programming language, which allows users to buy tickets online for movies in different cinemas. Online shopping has been growing in popularity in recent years and has become a common practice in many companies. There are several reasons why it was introduced:

**Convenience:** Online shopping allows customers to access the Internet from home, work or anywhere and easily purchase the products or services they want. They do not need to be in person at traditional stores or make an appointment.

**Wide selection:** In the online space, customers have access to a huge selection of products and services from many parts of the world. They are not limited to the inventory of local stores and can easily compare prices, brands and quality from the comfort of their own home.

**Time saving:** Online shopping is fast and efficient. Customers don't have to go through multiple stores or stand in line to pay. They can simply buy and pay for a product or service with a few clicks.

Because of these advantages, many companies decide to introduce or expand online shopping to reach and serve a wide range of online shoppers.

**Keywords:** *online, shopping, convenience.*

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>9</b>
<b>2. Projekt célja</b>	<b>10</b>
<b>3. Programok, technológiák bemutatása</b>	<b>11</b>
3.1. Beépített szoftverek, könyvtárak . . . . .	11
3.1.1. .NET Framework . . . . .	11
3.1.2. SQL - MySQL . . . . .	13
3.1.3. Docker . . . . .	14
<b>4. Szoftverek összehasonlítása</b>	<b>15</b>
4.1. Az én szoftverem és ezek a szoftverek közötti különbség . . . . .	17
<b>5. Rendszer specifikációja</b>	<b>18</b>
5.1. Felhasználói követelmények . . . . .	18
5.2. Use Case Diagram . . . . .	18
5.3. Rendszerkövetelmények . . . . .	19
5.3.1. Funkcionális követelmények . . . . .	19
5.3.2. Nem funkcionális követelmények . . . . .	20
<b>6. A Szoftver tervezése</b>	<b>21</b>
6.1. Arhitektúra . . . . .	21
6.1.1. "Front-end" arhitektúra . . . . .	21
6.1.2. Rendszer arhitektúra . . . . .	27
6.1.3. Back-end arhitektúra . . . . .	27
<b>7. UI Folyamat</b>	<b>31</b>
7.0.1. User . . . . .	32
7.0.2. Admin . . . . .	35
<b>Ábrák jegyzéke</b>	<b>40</b>
<b>Összefoglaló</b>	<b>41</b>
<b>Irodalomjegyzék</b>	<b>42</b>

# 1. fejezet

## Bevezető

Az internet fejlődése számos területen forradalmi változásokat hozott az ember életébe, beleértve az online vásárlást. Ezek az új lehetőségek a kényelmet és a hozzáférhetőséget hozták el az otthonainkba, megkönnyítve minden napjainkat és megváltoztatva az életünk módját.

Az internetes kereskedelem gyökerei egészen az 1960-as évekig nyúlnak vissza, amikor az Egyesült Államok kifejlesztették az ARPANET (Advanced Research Projects Agency Network) hálózatot. Az ARPANET fő célja az volt, hogy katonai kutatóintézeteket és egyetemeket kapcsoljon össze hálózatokon keresztül, lehetővé téve az adatcserét, például a számítási kapacitás megosztását. Az első online tranzakciók azonban csak az 1990-es években jelentek meg.

Az egyik példa az Amazon.com, amelyet Jeff Bezos alapított 1994-ben. Kezdetben az Amazon weboldaláról könyveket lehetett vásárolni, viszont az évek elteltével és az internet fejlődése során fokozatosan kiterjesztette a felhasználók számára, hogy más termékeket is kínáljon. Azóta számos más webhely és platform jött létre, amelyek lehetővé teszik az online vásárlást mint például az eBay, az eMag és számos más weboldal.

Az online vásárlás lehetőséget kínál számunkra, hogy kényelmesen, a saját otthonunkból vásárolhassunk. Csak egy internetkapcsolat és egy eszköz szükséges ahhoz, hogy hozzáférjünk a világ bármely részéről a kívánt termékünkhez. Nem szükséges többé időt töltenünk a fizikai jelenlétre a hagyományos üzletekben vagy hosszú sorokba állni a pénztárnál.

Az online vásárlás a mai világban rendkívül fontos szerepet játszik az ember életében, mivel lehetővé teszi, hogy bárhol és bármikor vásárolhassunk anélkül, hogy el kellene hagynunk az otthonunkat személyes vagy akár betegségi okokból. Az online vásárlás lehetővé teszi, hogy olyan termékeket vásároljunk, amelyek nem elérhetőek az országon belül.

Az online vásárlás során könnyedén hozzáférhetünk más vásárlók véleményeihez és termékértékeléseihez. Ez segíthet nekünk megbizonyosodni arról, hogy a választott termék minősége és teljesítménye megfelel-e az elvárásainknak.

Az online vásárlás egyre népszerűbbé válik, és folyamatosan fejlődik az ekereskedelem terén. Az emberek egyre inkább válasszák az online vásárlást a kényelem, a hozzáférés és az idő megtakarítása érdekében.

## **2. fejezet**

### **Projekt célja**

A projektem célja tehát a mozigegyek eladása, a filmszerető emberek számára online vásárláson keresztül, hogy kényelmesen és gyorsan foglalhassanak helyet kedvenc filmjeik vetítéseire a CinemaOne, CinePlexx és CinemaCity mozitermekben Romániában.

Az alkalmazásom lehetővé teszi a felhasználók számára, hogy böngésszenek a legfrissebb filmek között, megtekintsék a filmjegyek állapotát, rövid leírását, a film szereplőit, valamint egyszerűen megvásárolhassák a kiválasztott jegyeket online fizetési lehetőséggel Pay Pall-al.

A céлом az volt, hogy egy felhasználóbarát felületet tervezek és fejlesszék ki, amely lehetővé teszi a felhasználók számára a problémamentes jegyvásárlást, emellett pedig adatbiztonságra is nagy hangsúlyt fektettem az online tranzakciók során.

Így a web applikációm segítségével nemcsak mozitermeknek nyújtok egy hatékony rendszert, hanem a felhasználók számára is kellemes és kényelmes élményt biztosítok, amikor kedvenc filmjeiket szeretnék megnézni a legújabb technológiai fejlesztésekkel felvontató mozitermekben Romániában.

## 3. fejezet

# Programok, technológiák bemutatása

### 3.1. Beépített szoftverek, könyvtárak

A projektem elkészítéséhez Visual Studio-t használtam, ami egy szoftverfejlesztői környezet. Ezen belül .NET keretrendszerrel C# nyelvel írtam meg a kívánt eTickets című webalkalmazásomat. A projektemet dockerizáltam is, ami abban segít, hogy az alkalmazásom zavartalanul futhasson bármilyen környezetben. Az adatok eltárolásához a MySQL Workbench-et használtam. DBeaver adatbázis-kezelő eszközzel menedzseltettem, hogy könnyebben kezelhető legyen az adatok lekérdezése, illetve az adatok feltöltése.

#### 3.1.1. .NET Framework

A .NET egy olyan keretrendszer, amely szofverfejlesztésben segít. A .NET keretrendszer (a .NET Framework) különböző programozási nyelveken, például C#, Visual Basic.NET, F# vagy másokon keresztül lehetővé teszi a fejlesztők számára a Windows-alapú alkalmazások és webszolgáltatások készítését. A C# nyelvet használtam, mivel ezt a programozási nyelvet találták a legmegfelelőbb nyelvnek a .NET webalkalmazások előkészítéséhez.

A .NET keretrendszer számos alkalmazás területeken használatos, beleértve a webfejlesztést **ASP.NET**, asztali alkalmazásokat (WPF), adatbázis és szerveroldali alkalmazásokat és játékokat. Az én projektem ASP.NET webalkalmazás-fejlesztési keretrendszerben van megvalósítva.

#### ASP.NET

Az ASP.NET egy ingyenes webes keretrendszer nagyszerű webhelyek és webes alkalmazások készítéséhez HTML, CSS és JavaScript használatával. Webes API-kat is létrehozhat, és valós idejű technológiákat használhat, mint például a Web Sockets. Az ASP.NET három keretrendszert nyújt a webalkalmazások létrehozásához: Web Forms, **ASP.NET MVC** és ASP.NET Web Pages. Mindhárom keretrendszerrel nagyszerű webes alkalmazásokat hozhatunk létre. Nem számít, milyen keretrendszert választ, az ASP.NET minden előnyét és funkcióját mindenhol megkapja.

Mindegyik keretrendszer más-más fejlesztési stílust céloz meg. A választott programozási eszközök (tudás, készségek és fejlesztési tapasztalat), a létrehozandó alkalmazás típusa és az Ön számára megfelelő fejlesztési megközelítés kombinációjától függ.

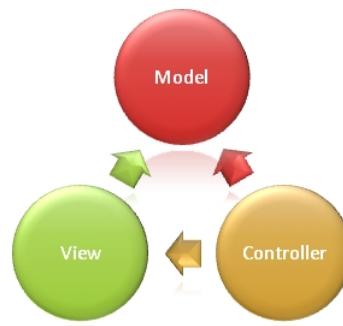
## ASP.NET MVC

Az én projektemhez ASP.NET MVC-vel valósítottam meg. Az ASP.NET MVC hatékony, mintákon alapuló módszert kínál dinamikus webhelyek létrehozására, amely lehetővé teszi a problémák tiszta szétválasztását, és teljes ellenőrzést biztosít a jelölés felett az élvezetes, agilis fejlesztés érdekében.

### MVC

Az MVC ([3.1 ábra](#)) keretrendszer a következő fő komponensekből áll:

1. Model (Modell): Ez képviseli a a webalkalmazás adattartalmát és az üzleti logikáját. Itt találhatók az adatstruktúrák, az adatbáziskapcsolatok és az üzleti logika megvalósítása.
2. View (Nézet): Ez felelős az adatok megjelenítéséért a felhasználók számára. A nézetek HTML-alapú sablonok.
3. Controller (Vezérlő): Ez közvetíti az adatokat a View és a Model között. Ez első lépésként fogadja a felhasználóktól a kéréseket, ez után pedig kezeli az eseményeket és végül pedig végrehajtják az adatok feldolgozását.



**3.1. ábra.** MVC

### 3.1.2. SQL - MySQL

Az adatok kezelésére és az információk adatbázisban való eltárolásához SQL-t használtam. Az adatokat **MySQL Workbench**-ben ellenőriztem és **DBeaver** segítségével menedzseltem.

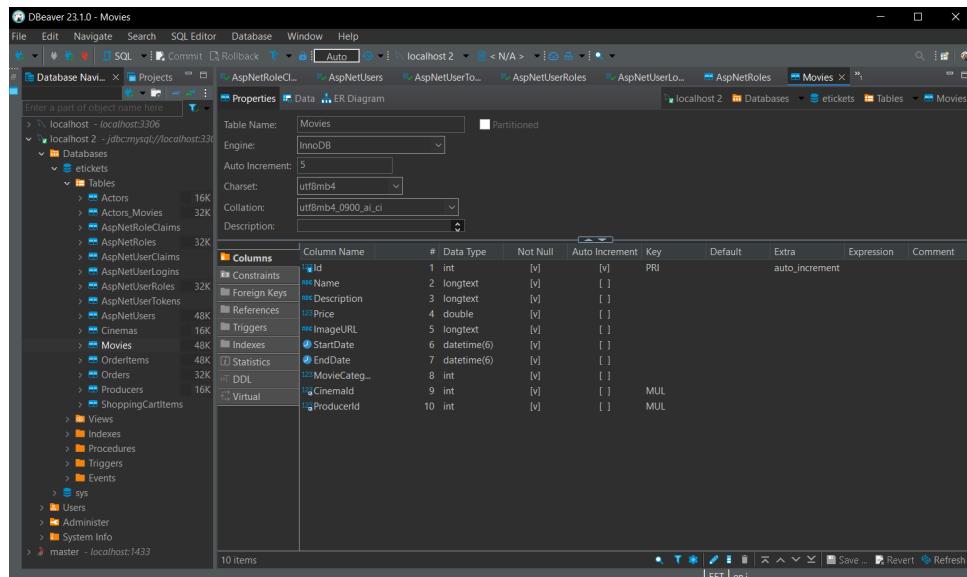
#### MySQL Workbench

A olyan funkciókkal rendelkezik, amelyek megkönnyítik az adatbázis-tervezést, az adatbázis-sémák létrehozását és módosítását, valamint az adatok kezelését. A projekt-hez szükségem volt használni a MySQL Workbench-et, mivel ez tartalmaz egy olyan beépített funkciót, amely lehetővé teszi a MySQL adatbázis szerver telepítését és konfigurálását. Ez a funkció a "MySQL Server" része a MySQL Workbench-nek.

#### DBeaver

A DBeaver adatbázis-kezelő eszköz, és nem tartalmaz beépített funkciót a szerver telepítésére vagy konfigurálására. A DBeaver használata előnyösebb, mint a MySQL Workbench, mivel több adatbázis-kezelő rendszert támogat, mint például PostgreSQL, Oracle, SQL Server MongoDB és sok más. DBeaver számos hasznos funkciót kínál az adatbázis-kezeléséhez, például adatimport és export, SQL szerkesztő és végrehajtás, adatellenőrzés és sok más.

A DBeaver felülete (3.2 ábra):



3.2. ábra. DBeaver

### 3.1.3. Docker

A Docker egy olyan platform, amely lehetővé teszi a szoftverek konténizált futtatását. A dockerizálást a "Docker Desktop" alkalmazás használatával oldottam meg.

A konténerizáció lehetővé teszi, hogy az alkalmazások fussenak, függetlenül attól, hogy milyen operációs rendszert használ az illető személy.

A Dockernek számos előnye van, amelyek közé tartozik:

1. Hordozhatóság: a "container"-ek örök környezetet biztosítanak, így az alkalmazásokat bármely rendszeren futtathatjuk.
2. Gyors telepítés és elindítás: A "container"-ek gyorsan telepíthetők és elindíthatóak, mivel nem szükséges operációs rendszert vagy virtuális gépet létrehozni.

#### Docker működése

A Docker működése a következő lépésekkel áll:

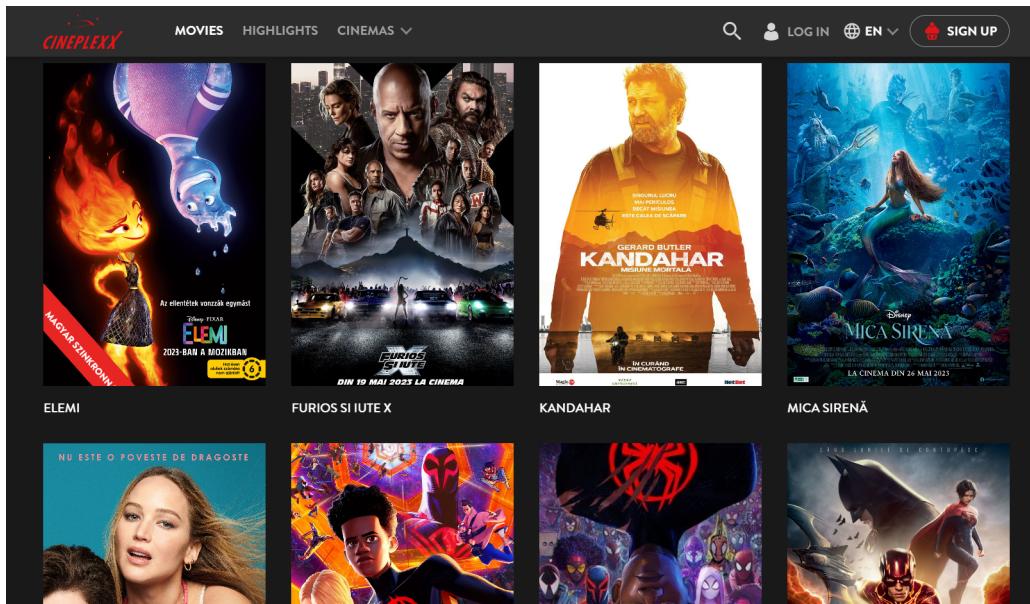
1. Image (kép) létrehozása: ez tartalmazza a webapplikációhoz tartozó függőségek futtatható állapotát.
2. Container: ez elkülönült környezetet biztosít a webapplikációnak, beleértve a fájlrendszerét, hálózati interfészt és erőforrásokat.
3. Dockerfile: ez egy egyszerű szöveges fájl, amelyben leírjuk, hogy milyen lépésekkel kell összeállítani a képet. Például meghatározzuk, hogy melyek fájlokat másolunk be és milyen parancsokat futtatunk. Ezt szimplán Visual Studio-ban létrehozhatjuk ha letöltöttük a Docker Desktop-ot.
4. Rendszerizoláció: A Docker használja a Linux kernel funkcióit a rendszerizolációhoz, így a konténerek elkülönítettek egymástól és a gazdagéptől. Ez biztosítja a konténerek izolációját és biztonságát.
5. Verziókezelés: A Docker lehetővé teszi az image-ok és konténerek verziókezelését, így könnyen visszaállíthatunk egy korábbi verzióra vagy frissíthetjük az alkalmazást az újabb verzióra.
6. Image megosztása: Az image-okat megoszthatjuk egy központi Docker Image Registry-ben, min például a Docker Hub. Ez lehetővé teszi másoknak, hogy könnyen hozzáférjenek az elkészült image-okhoz, és azokból konténereket indítsanak.

A fenti lépések segítségével a Docker lehetővé teszi az alkalmazások egyszerű telepítését, futtatását és skálázását konténerekben, biztosítva a konzisztenciát, hordozhatóságát és rendszerizolációt.

## 4. fejezet

### Szoftverek összehasonlítása

Akik Marosvásárhelyen laknak, azok halottak a Cineplexx moziterméről. Ennek a moziteremnek van egy saját weboldala is, ahonnan mozigyereket lehet rendelni. Itt is vannak olyan funkciók, amelyek léteznek az általam megvalósított szoftveremben. (Link az oldalra: [Cineplexx](#))



4.1. ábra. Cineplexx

Romániában van a Cinema One nevezetű moziterem is, ami pontosabban Brassó városában található. Ennek is van egy saját weboldala, aminek a célja pont ugyanaz, mint a Cineplexx moziterem oldalának. (Link az oldalra: [CinemaOne](#))



ACUM LA CINEMA ONE

ÎN CURÂND

<span style="font-size: 1.5em;">&lt;</span> <b>IUN</b> <b>24</b> Sam <span style="font-size: 0.8em;">▼</span>	<b>IUN</b> <b>25</b> Dum <span style="font-size: 0.8em;">▼</span>	<b>IUN</b> <b>26</b> Lun <span style="font-size: 0.8em;">▼</span>	<b>IUN</b> <b>27</b> Mar <span style="font-size: 0.8em;">▼</span>	<b>IUN</b> <b>28</b> Mie <span style="font-size: 0.8em;">▼</span>	<b>IUN</b> <b>29</b> Joi <span style="font-size: 0.8em;">▼</span>	<b>AUG</b> <b>19</b> Sam <span style="font-size: 0.8em;">▼</span>	<span style="font-size: 1.5em;">&gt;</span>	<b>ALEGE CÂND</b> Vezi ce filme rulează pe zile	
--	--	--	--	--	--	--	---	--	--

#### 4.2. ábra. Cinema One

Romániában még van egy moziterem, ami már több városban elérhető. Ennek a neve a Cinema City. (Link az oldalra: [CinemaCity](#))

**CINEMA CITY**

Alege cinematograful

RO ▾

PROGRAM
OFERTĂ
VOUCHERE CADOU
BLOG
IMAX
4DX
VIP
EDUCATIONAL

ACUM ÎN PROGRAM

Metallica M72 World Tour Live, Seară a Două

Metallica M72 World Tour Live, Seară Intâi

Ruby Gillman, adolescente kraken

Indiana Jones și Cadranul Destinului Sub

Mi-e silă de mine

Întâlnirile VORBITORINCII

ÎN CURÂND

#### 4.3. ábra. Cinema City

## **4.1. Az én szoftverem és ezek a szoftverek közötti különbség**

Az én szoftverem és a korábban említett szoftverek között a különbség az, hogy míg ezek a szoftverek csak egyetlen moziterem filmjeit teszik elérhetővé, az én szoftverem az összes moziterem filmjeinek a jegyét elérhetővé teszi számunkra. Ez azt jelenti, hogy az én szoftverem sokkal átfogóbb és változatosabb filmkínálatot biztosít, amely kielégítheti a különböző ízléseket és preferenciákat.

Ezenkívül fontos megjegyezni, hogy míg a korábban említett szoftverek letölthetőek a Google Play telefonos applikációról Android operációs rendszerű telefonokra vagy az Apple Store-ról iOS operációs rendszerű telefonokra, az én szoftverem nem tölthető le mobil telefonokra.

Mindegyik szoftver különböző szoftverfejlesztői környezetben van megírva, illetve a front-end-hez külön szoftverfejlesztési környezetet használnak mint például a Vue.js-t. Ez lehetővé teszi a fejlesztőknek a hatékony és rugalmas webalkalmazások készítését. (A front-end felelős a webalkalmazások kinézetéért de erre még kitérünk).

Az én szoftverem kinézetét azonban Visual Studio fejlesztői környezetben készítettem el.

## 5. fejezet

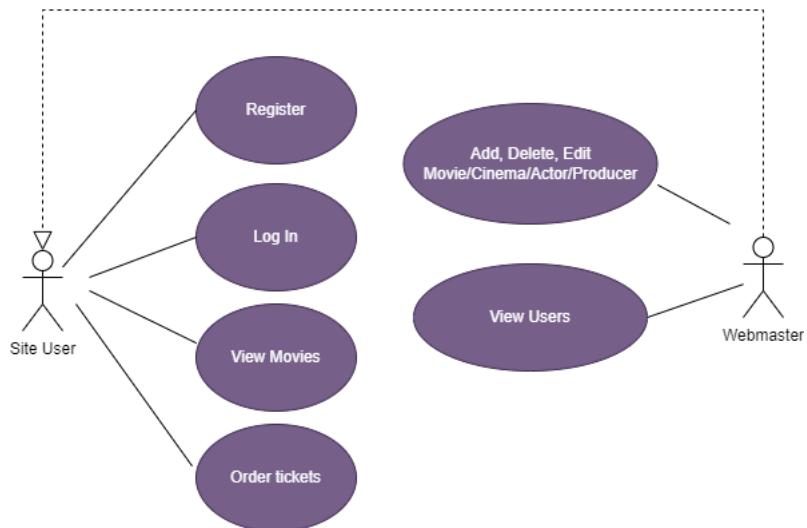
# Rendszer specifikációja

### 5.1. Felhasználói követelmények

A felhasználó amint megnyitja a weboldalt a Főoldalt látja elsőként. A Főoldalon megjelennek a filmek, amik kártya szerűen vannak elrendezve. Ezekben a "kártyákon" megvan jelenítve a film címe, a film képe és az ára.

Ahhoz, hogy a felhasználó tudjon filmjegyeket rendelni, kell regisztrálnjon a weboldalon. A regisztráláshoz kell a felhasználó teljes neve, email címe és a bejelentkezéshez használt titkos jelszó. A jelszónak tartalmaznia kell legalább egy nagy karaktert, egy kis karaktert, egy számjegyet, egy speciális karaktert és legalább 8 legyen a karakterlánc összehossza.

### 5.2. Use Case Diagram



5.1. ábra. Use Case Diagram

A weboldalra két felhasználóként tudsz bejelentkezni: adminisztrátorként(admin) és kliensként(user).

- Ha **szimpla felhasználó/kliens**-ként jelentkezel be a weboldalra, akkor elérheted a következő ablakokat/oldalakat: filmek, filmek leírás, színészek, szervezők. Ha viszont regisztrálsz a weboldalra és utánna bejelentkezel, akkor a megrendelési oldal is elérhető lesz.
- Ha pedig **adminisztrátor**-ként jelentkezel be, akkor minden oldal elérhetővé válik számodra még annyi előnyvel, hogy tudsz filmeket/mozitermeket/színészeket/szervezőket hozzáadni, módosítani, illetve törülni.

## 5.3. Rendszerkövetelmények

### 5.3.1. Funkcionális követelmények

- A weboldal megnyitása következtében megjelenik a Főoldal, valamint a filmek, amik "kártya" szerűen vannak elrendezve.
- A Főoldalon a felhasználó megnézheti, hogy milyen filmek elérhetőek, milyen árban vannak ezek és egy kis leírást a filmkről.
- A jobb felső sarokban megjelenik két darab gomb, a "Register" és a "Log in" gombok, illetve a kereső mező.
- Ha a felhasználó beírja a kereső mezőjébe a filmnek a pontos címét és megnyomja a nagyító ikont, akkor csak azt az egy filmet fogja kimutatni a Főoldal.
- A felhasználó megtudja nézni a film rövid leírását, ha a "Show Details" gombra kattint.
- Ha a filmnek a "Status"-nál (Állapot) azt írja pirossal, hogy "Expired", akkor nem lehet rákattintani az "Add to Cart" (tedd a kosárba) gombra.
- Ha viszont a "Status"-a "Available" (elérhető), akkor rálehet kattintani az "Add to Cart" gombra, viszont a Log In ablak nyitódik ki.
- Amint a felhasználó rákattint a "Register" gombra, egy újabb ablak nyitódik ki. Ezen az ablakon a felhasználó regisztrálni tud az oldalra.
- Miután a regisztrálás sikeresen teljesült, a felhasználó meglássa a sikeresen teljesített szöveget.
- Ha a felhasználó a regisztrálás után rákattint a bal felső sarokba az "eTickets" szövegre, akkor a Főoldal nyitódik ki újra.
- Ha a felhasználó rákattint a "Log in" (bejelentkezés) gombra és kitölti a mezőket, akkor elérhetővé válik számára a filmjegyek megrendelése!
- Jobb felső sarokból eltűntek a "Register" és a "Log in" gombok, viszont megjelent a "Log out" gomb.

- A felhasználó kitud lépni a fiókjából, ha a "Log Out" gombot megnyomja.
- A megrendelés csak akkor lehetséges, ha a filmnél a "Status"-a "Available" (elérhető) vagy "Upcoming" (közelgő).
- Miután a felhasználó rákattint a kosárba adás gombra, kinyílik egy újabb oldal. Ezen az oldalon megtekinthető az összes jegy megrendelése. A zöld kosár icon kattintásával megnő a filmjegy ára és darabszáma. A piros kosár icon pedig törli a filmjegyet.
- Amint a felhasználó rákattint a kosár adás gombra vagy már létezik filmjegy a kosárban, akkor a "Log out" gomb mellett megjelenik egy kosár icon illetve egy szám, ami a filmjegyek számát mutassa.
- A megrendelés Pay Pal segítségével történik.

### **5.3.2. Nem funkcionális követelmények**

A weboldal megfelelő működéséhez a következő rendszerkövetelményeknek kell eleget tenni:

#### **Web alkalmazás**

- Laptop/Számítógép
- Windows 10/11 alapú operációs rendszer
- Képernyő mérete: 1920x1080
- Egér/Laptop érintőfelület
- Billentyűzet
- Internet hozzáférés
- Böngésző

#### **Server futtatása**

- .NET (verzió 6.0)
- Docker Desktop (verzió 20.10.20)
- Visual Studio 2022
- MySQL Workbench (verzió 8.0)
- Internet hozzáférés

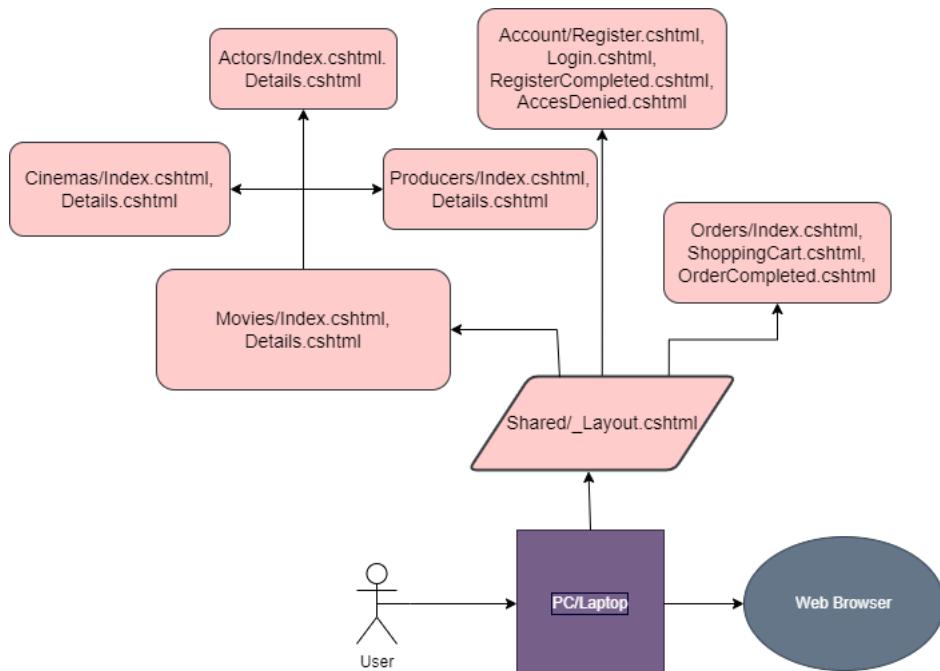
# 6. fejezet

## A Szoftver tervezése

A tervezés folyamat elején eldőlt, hogy az eTickets weboldalon milyen Modellek (táblázatok) létezzenek az adatbázisban. Hogyha már az alap ötleteim megvoltak, szükségem volt programozási nyelvekre, ahol meg is tudjam valósítani azokat. A következő programozási nyelveket használtam: C#, ASP.NET, HTML, CSS. A frissen létrehozott projektemet egy **GitHub repository**-hoz csatoltam, ahova feltöltöttem a haladásomat. (link: [eTickets](#))

### 6.1. Arhitektúra

#### 6.1.1. "Front-end" arhitektúra



6.1. ábra. Front-end arhitektúra

A "front-end" felelős a weboldal kinézetéért. A felhasználó egy számítógép segítségével eléri a weboldalt. Az architektúra diagramon (lásd 6.1 ábra) a következők kerülnek bemutatásra:

- Shared/\_Layout.cshtml: ez egy legfelső szintű sablont határoz meg az alkalmazás nézeteihez. Az alkalmazások nem igényelnek elrendezést. (Link: [\\_Layout.cshtml](#))
- Movies/Index.cshtml:

### 6.1. kódrészlet. Filmek oldala

```
@model IEnumerable<Movie>

{@
    ViewData["Title"] = "List of Movies";
}
```

Ez kód részlet lehetővé teszi a sorozat elemeken való végig iterálását és használatával olyan műveleteket végezhetünk vele, mint például a filmek végig iterálása foreach ciklussal.

### 6.2. kódrészlet. Filmek oldal

```
@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))
{
    <a class="text-white float-right" asp-action="Edit"
        asp-route-id="@item.Id"><i class="bi bi-pencil-square"></i></a>
}
```

Ezzel a kód részlettel megjelenítettem az admin típusú felhasználónak az "Edit" (Módosít) gombot a film "kártyán", ami ecset icon-ként van megjelenítve. Ha arra rákattint az admin, akkor megnyílik a módosító ablak, ahol az admin tudja módosítani a film adatait.

### 6.3. kódrészlet. Filmek oldala

```
<div class="col-md-6">
    
</div>
<div class="col-md-6">
    <div class="card-body">
        <p class="card-text">@item.Description</p>
        <p class="card-text"><b>Cinema:</b>
            </b>@item.Cinema.Name</p>
        <p class="card-text"><b>Category:</b> @item.MovieCategory</p>
        <p class="card-text"><b>Start Date:</b>
            </b>@item.StartDate.ToString("dd MMM yy")</p>
        <p class="card-text"><b>End Date:</b> @item.EndDate.ToString("dd
            MMM yy")</p>
        <p class="card-text">
```

```

<b>Status: </b>
@if (DateTime.Now >= item.StartDate && DateTime.Now <= item.EndDate)
{
    <span class="badge bg-success text-white">AVAILABLE</span>
}
else if (DateTime.Now > item.EndDate)
{
    <span class="badge bg-danger text-white">EXPIRED</span>
}
else
{
    <span class="badge bg-primary text-white">UPCOMING</span>
}
</p>
</div>
</div>

```

Ebben a kódrészletben található az, hogy hogyan is néz ki maga a film "kártya", illetve a film állapota.

#### 6.4. kódrészlet. Filmek oldala

```

<a class="btn btn-outline-primary float-right" asp-action="Details"
    asp-route-id="@item.Id">
    <i class="bi bi-eye-fill"></i> Show Details</a>
<a class="btn btn-success text-white @((DateTime.Now > item.EndDate) ?
    "disabled" : "")"
asp-controller="Orders"
asp-action="AddItemToShoppingCart"
asp-route-id="@item.Id">
    <i class="bi bi-cart-plus"></i> Add to Cart (Price
        @item.Price.ToString("c"))</a>

```

Itt találhatóak a gombok "Show Details" és az "Add to Cart". Az asp-controller="Orders" asp-action="AddItemToShoppingCart" attribútum beállítja a hivatkozás célját az "Orders" vezérlőre, az "AddItemToShoppingCart" akciótól. Amikor a hivatkozásra kattintanak, az alkalmazás az "AddItemToShoppingCart" akciót hajtja végre a "Orders" vezérlőben.

- Movies/Details.cshtml: Ez csak akkor érheti el a felhasználó ha a film "kártyán" lévő "Show Details" gombra kattint.
- Producers/Details.cshtml: Ezt is csak akkor érheti el a felhasználó, ha a film "kártyán" lévő "Show Details" gombra kattint. Ez után pedig rákattint a "Producer" melletti névre.
- Producers/Index.cshtml: Ezt is csak akkor érheti el a felhasználó, ha a film "kártyán" lévő "Show Details" gombra kattint. Ez után pedig rákattint a "Producer" melletti névre és ez után pedig a "Show all" gombra kattint.

- Cinemas/Index.html, Details.html és Actors/Index.html, Details.html: Ezek is pont úgy működnek mint a Producers, azzal a különbséggel, hogy más gombokra kell kattintani.
- Account/Register.cshtml:

#### 6.5. kódrészlet. Regisztrációs oldal

---

```

<form asp-action="Register">
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
<div class="form-group">
    <label asp-for="FullName" class="control-label"></label>
    <input asp-for="FullName" class="form-control" />
    <span asp-validation-for="FullName" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="EmailAddress" class="control-label"></label>
    <input asp-for="EmailAddress" class="form-control" />
    <span asp-validation-for="EmailAddress"
        class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Password" class="control-label"></label>
    <input asp-for="Password" class="form-control" />
    <span asp-validation-for="Password" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="ConfirmPassword" class="control-label"></label>
    <input asp-for="ConfirmPassword" class="form-control" />
    <span asp-validation-for="ConfirmPassword"
        class="text-danger"></span>
</div>

<div class="form-group">
    <input class="btn btn-outline-success float-right" type="submit"
        value="Sign up" />
    <a class="btn btn-outline-secondary" asp-controller="Movies"
        asp-action="Index">Cancel</a>
</div>
</form>

```

---

A regisztrációt egy form alá tettek. Ebben az űrlapban négy mezőt (FullName, EmailAddress, Password és ConfirmPassword) hoztam létre. Az "asp-for" attribútum beállítja a mezőkhöz tartozó adatmezőket, így az űrlap beküldésekor az értékek automatikusan bekerülnek a modell tulajdonságaiiba.

- Orders/ShoppingCart.cshtml:

#### 6.6. kódrészlet. Bevásárló kosár oldal

---

```
<table class="table">
```

---

```

<thead>
    <tr>
        <th>Selected amount</th>
        <th>Movie</th>
        <th>Price</th>
        <th>Subtotal</th>
        <th></th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model.ShoppingCart.ShoppingCartItems)
    {
        <tr>
            <td class="align-middle">@item.Amount</td>
            <td class="align-middle">@item.Movie.Name</td>
            <td
                class="align-middle">@item.Movie.Price.ToString("c")</td>
            <td class="align-middle">@((item.Amount *
                item.Movie.Price).ToString("c"))</td>
            <td class="align-middle">
                <a class="text-danger" asp-controller="Orders"
                    asp-action="RemoveItemFromShoppingCart"
                    asp-route-id="@item.Movie.Id">
                    <i class="bi bi-cart-dash"></i>
                </a>
                |
                <a class="text-success" asp-controller="Orders"
                    asp-action="AddItemToShoppingCart"
                    asp-route-id="@item.Movie.Id">
                    <i class="bi bi-cart-plus"></i>
                </a>
            </td>
        </tr>
    }
</tbody>
<tfoot>
    <tr>
        <td colspan="2"></td>
        <td><b>Total:</b></td>
        <td>
            @Model.ShoppingCartTotal.ToString("c")
        </td>
        <td></td>
    </tr>
</tfoot>
</table>

```

A fenti kódrészlet egy táblázatot jelenít meg. A "Selected amount" (Kiválasztott mennyiség) a kiválasztott mennyiséget jeleníti meg a bevásárlókosárban lévő ele-

mekhez. A "Movie" a film nevét, a "Price" a film árát és a "Subtotal" a részösszeget jeleníti meg amely a darabszám és az ár szorzatának eredménye. Ez pénzügyi formátumba van megjelenítve.

A "foreach" ciklus segítségével vegig iteráltam a "Model.ShoppingCart.ShoppingCartItmes" kollekciót, és minden elemhez generáltam egy "<tr>" elemet a táblázatban.

A "<a>" elemek a "RemoveItemFromShoppingCart" és "AddItemToShoppingCart" akcióhoz tartozó linkekkel hoznak létre. Ezekkel az akciókkal a bevásárlókosár elemeit lehet törülni vagy hozzáadni a kosárhoz.

A "<tfoot>" elem a táblázat láblécét tartalmazza. Ez a lábléc az oszlopcsoporthoz tartozó összes elem részösszegét jeleníti meg a "Model.ShoppingCartTotal" segítségével.

- Orders/OrdersCompleted.cshtml:

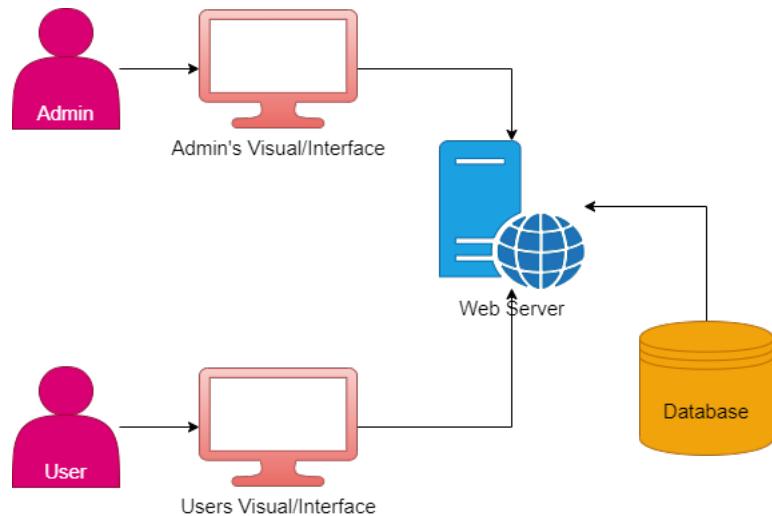
#### 6.7. kód részlet. Sikeres megrendelési oldal

```
<div class="row">
    <div class="col-md-6 offset-3 alert alert-success text-center">
        <h2>Order completed successfully</h2>
        <p>You can check all your orders in the Orders section of your profile</p>
        <hr />
        <p>Thank you!</p>
    </div>
</div>
```

Ez megjeleníti "<h2>" és a "<p>"-ben lévő szöveget, miután a rendelés sikeresen teljesült a felhasználó által.

A fenti kód fájlok a "View" mappa alatt találhatóak.

### 6.1.2. Rendszer arhitektúra



6.2. ábra. Rendszer arhitektúra

### 6.1.3. Back-end arhitektúra

A modellek adatbázistáblákkal vannak összekapcsolva. Ezek a táblák adatokat tárolnak, mint például felhasználók és termékek. Az én adatbázisomban levő termékek közé tartoznak a filmjegyekhez kapcsolódó információk: színészek, szervezők, mozitermek stb.

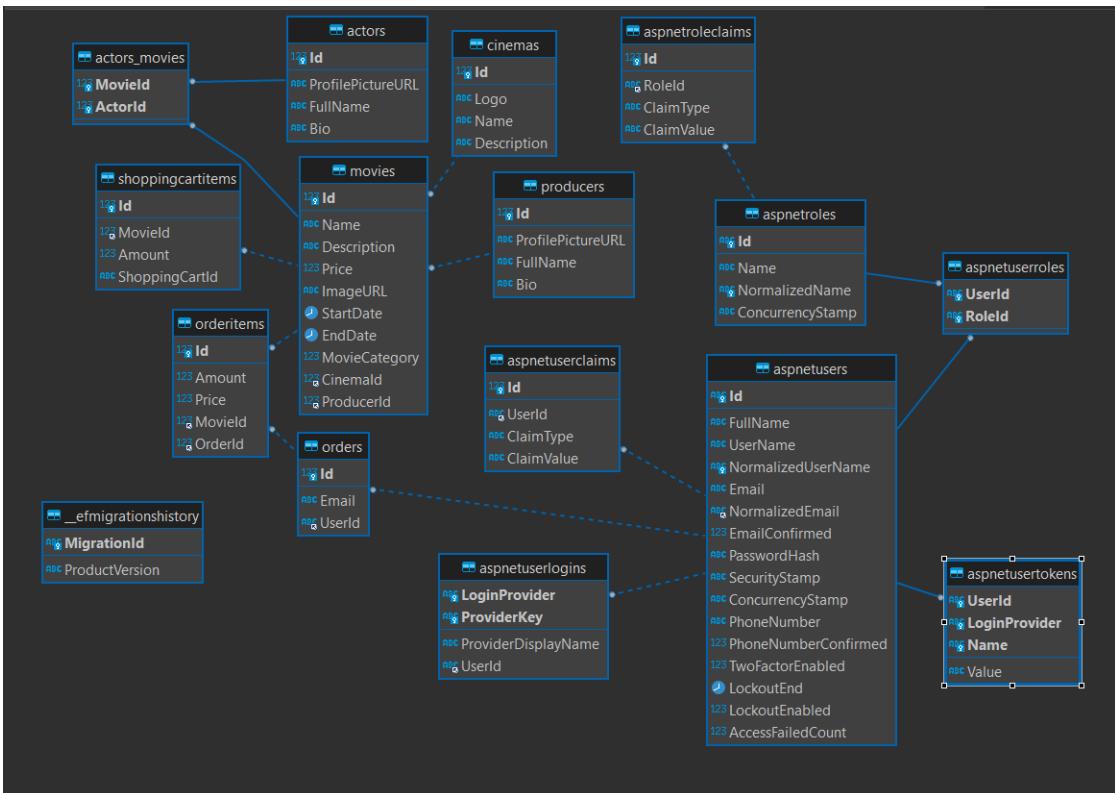
A projektem sikeres elkészítéséhez, **először** létrehoztam a következő modelleket: Actor, Producer, Cinema, Movie.

Miután ezeket létrehoztam, a projektemhez hozzáadtam egy DbContext fájlt, ami egy fordító a modellosztályai és az adatbázis között. (6.3 ábra)



6.3. ábra. DbContext

A következő táblákat hoztam létre migrálás során (ábra 6.4):



**6.4. ábra.** Adatbázis

A migrálás feladata az, hogy szinkronizálja a modell változásait az adatbázissémaival. A migrálás "Visual Studio"-ban a Package Manager Console-ban történik a következő kódrészlettel:

#### 6.8. kódrészlet. Migráció

```
PM> Add-Migration Initial
PM> Update-Database
```

Ahhoz hogy tudjunk migrálni, lekell tölteni a "Microsoft.EntityFrameworkCore.Tools" csomagot.

A controller felelős a front-end számára elérhető végpontok (endpoints) kialakításáért, amelyeken keresztül a HTTP kérésekkel kommunikál. Ez a réteg végzi az adatokát átadását a front-end és a szolgáltatások között. A végpontok lehetővé teszik a front-end számára, hogy kéréseket küldjön a szerver felé és válaszokat kapjon, például adatok lekérdezésére, adatok beküldésére vagy műveletek végrehajtására.

**Második** lépésként létrehoztam a modellekhez tartozó controllereket.

**Harmadik** lépében miután a kinézetét megírtam a modelleknek, a szervizekkel foglalkoztam.

A szervizek olyan metódusokat valósítanak meg, amelyekre szükség van a vezérlőnek. Ezek a módszerek végzik az üzleti logika végrehajtását, például adatok lekérdezését vagy módosítását az adatbázisban.

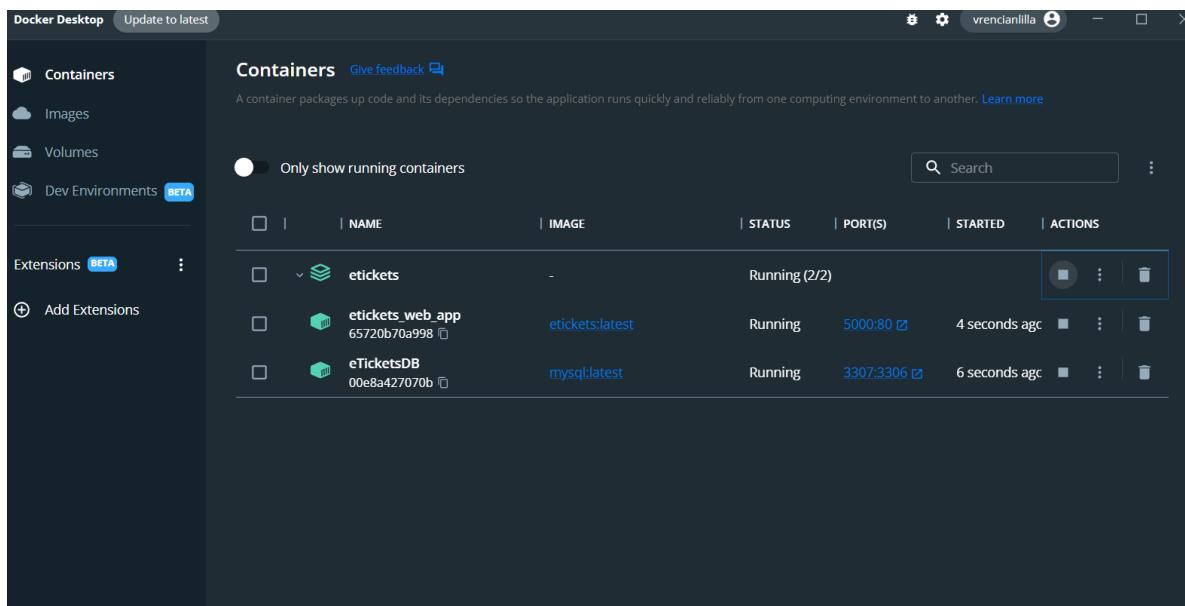
## Dockerizálás

A dockerizálást Visual Studio-ban úgy oldottam meg, hogy létrehoztam egy docker fájlt Visual Studio-ban és a következő kódrészletet írtam belé:

### 6.9. kódrészlet. docker-compose.yml

```
services:  
  web_app:  
    build: .  
    image: etickets  
    container_name: etickets_web_app  
    ports:  
      - "5000:80"  
    depends_on:  
      - db  
  db:  
    image: "mysql"  
    container_name: eTicketsDB  
    restart: always  
    ports:  
      - "3307:3306"  
  environment:  
    MYSQL_DATABASE: "etickets"  
    MYSQL_USER: user1  
    MYSQL_PASSWORD: Taska189*  
    MYSQL_ROOT_PASSWORD: Taska189*
```

Ez kódrészlet létrehozta a Docker Desktop alkalmazásba a 6.5 képen található container-eket.



6.5. ábra. Docker Desktop

Bontsuk le a kódot szakaszonként:

```
services:
```

Ez a szakasz az alkalmazást alkotó tárolókat határozza meg.

```
web_app:  
build: .  
image: etickets  
container_name: etickets_web_app  
ports:  
- "5000:80"  
depends_on:  
- db
```

Itt definiálunk egy "web\_app" nevű szolgáltatást. Az aktuális könyvtár (.) felhasználásával készült összeállítási környezetként. Az eredményül kapott kép "jegyek" címkével lesz ellátva. A tároló neve „etickets\_web\_app” lesz. A szolgáltatás a gazdagép 5000-es portjára és a tároló 80-as portjára van leképezve ("5000:80"). Ez a "db" nevű szolgáltatástól függ, ami azt jelenti, hogy a "db" szolgáltatás a "web\_app" szolgáltatás előtt indul el.

```
db:  
image: "mysql"  
container_name: eTicketsDB  
restart: always  
ports:  
- "3307:3306"  
environment:  
MYSQL_DATABASE: "etickets"  
MYSQL_USER: user1  
MYSQL_PASSWORD: Taska189*  
MYSQL_ROOT_PASSWORD: Taska189*
```

Ez a szakasz egy másik "db" nevű szolgáltatást határoz meg. A Docker Hub "mysql" képét használja. A tároló neve „eTicketsDB”. Az újraindítás: minden opció biztosítja, hogy a tároló automatikusan újra induljon, ha leáll. A gazdagép 3307-es portja a tároló 3306-os portjához van hozzárendelve ("3307:3306"). Környezeti változók vannak beállítva a MySQL tárolóhoz, beleértve az adatbázis nevét (MYSQL\_DATABASE: "etickets"), a MySQL felhasználót (MYSQL\_USER: user1), a felhasználó jelszavát (MYSQL\_PASSWORD: Taska189\*) és a root jelszót (MYSQL\_ROOT\_PASSWORD: Taska189\*).

Összefoglalva, ez a Docker Compose konfiguráció egy többtárolóból álló alkalmazást határoz meg, amely két szolgáltatásból áll: egy webalkalmazásból (web\_app) és egy MySQL adatbázisból (db). A webalkalmazás-szolgáltatás az aktuális könyvtárból épül fel, a gazdagép 5000-es portjára van leképezve, és az adatbázis-szolgáltatástól függ. Az adatbázis-szolgáltatás a MySQL lemezket használja, leképezi a 3307-es portot a gazdagépen, és környezeti változókat állít be a MySQL-tárolóhoz.

# 7. fejezet

## UI Folyamat

A 7.1 ábrán látható a Főoldal, amin található egy "Log In" gomb és egy "Register" gomb a jobb felső sarokban, egy "Add To Cart" gomb és egy "Show Details" gomb a film kártyákon, valamint két Logo amik a bal felső sarokban találhatóak. Mind a két Logo a 7.1 ábrához vezet, ha a felhasználó rájuk kattint.

A "Show Details" gomb a 7.2 oldalhoz vezet, ami a film leírását mutassa. Az "Add To Cart" gomb a 7.3 oldalhoz vezet, valamint a "Log In" oldalhoz ha még nem fordult elő bejelentkezés.

The screenshot shows a movie listing interface. At the top right is a search bar with placeholder text "Search for a movie...", a magnifying glass icon, a "Log in" button, and a "Register" button. Below the search bar are four movie cards:

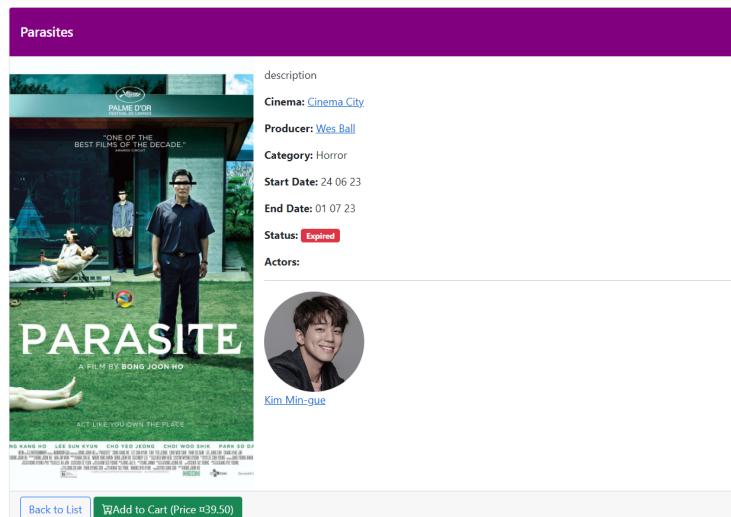
- The Maze Runner**: This card features a movie poster for "The Maze Runner" and includes the following details:
  - This is the Maze Runner description
  - Cinema: Cinema OneE
  - Category: science\_fiction
  - Start Date: 30 Jun 23
  - End Date: 03 Jul 23
  - Status: AVAILABLE

Buttons at the bottom: "Show Details" and "Add to Cart (Price #29.50)".
- Parasites**: This card features a movie poster for "Parasite" and includes the following details:
  - This is Parasites description
  - Cinema: Cinema City
  - Category: Horror
  - Start Date: 24 Jun 23
  - End Date: 01 Jul 23
  - Status: EXPIRED

Buttons at the bottom: "Show Details" and "Add to Cart (Price #39.50)".
- The Walk**: This card features a movie poster for "The Walk" and includes the following details:
  - This is The Walk description
  - Cinema: Cineplex
  - Category: Drama
  - Start Date: 14 Jun 23
  - End Date: 04 Jul 23
  - Status: AVAILABLE

Buttons at the bottom: "Show Details" and "Add to Cart (Price #39.50)".
- Hotel Transylvania 4**: This card features a movie poster for "Hotel Transylvania 4" and includes the following details:
  - This is Hotel Transylvania 4 movie description
  - Cinema: Cineplex

7.1. ábra. Főoldal/Filmek



**7.2. ábra.** Részletek

Ha 7.2 ábrán látható "Back To List" gombra kattintunk, akkor visszatérünk a 7.1 ábrához.

A következő oldalon megjelenik a bejelentkezésre szolgáló form:

Log in to your account

Email address	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Log in"/>	<input type="button" value="Cancel"/>

**7.3. ábra.** Bejelentkezés

## 7.0.1. User

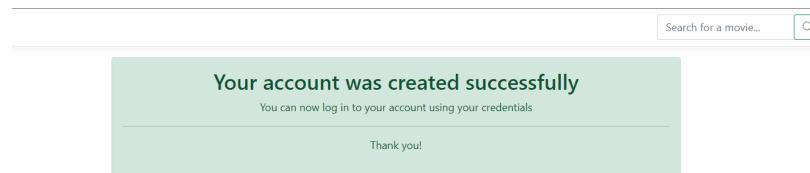
A "Register" gomb kattintása után a 7.4 ábrát lássuk. Ezen az oldalon történik a regisztrálás a weboldalhoz.

Sign up for a new account

Full name	<input type="text"/>
Email address	<input type="text"/>
Password	<input type="password"/>
Confirm password	<input type="password"/>
<input type="button" value="Sign up"/>	<input type="button" value="Cancel"/>

**7.4. ábra.** Regisztrálás

Ha sikeres a regisztrálás, akkor a következő oldalt lássuk:



**7.5. ábra.** Sikeresen létrehozva

Ha a regisztrálás sikerült és megnyomjuk a "Log In" gombot, akkor a **7.3** ábrát lássuk. Ha ezen a oldalon is kitöltsük az oszlopokat és rákattintunk a "Log In" gombra, akkor a **7.6** ábrát lássuk.

A screenshot of a movie listing page. At the top left is a navigation bar with "eTickets" and "Movies" tabs, and a search bar at the top right with placeholder text "Search for a movie...", a magnifying glass icon, and a "Log out" button. The main content area displays four movie cards: 1. "The Maze Runner" (Cinema OneE, science\_fiction, Jun 30 - Jul 03, Available) with "Show Details" and "Add to Cart (Price ₪29.50)" buttons. 2. "Parasites" (Cinema City, Horror, Jun 24 - Jul 01, EXPIRED) with "Show Details" and "Add to Cart (Price ₪39.50)" buttons. 3. "The Walk" (Cineplex, Drama, Jun 14 - Jul 04, Available) with "Show Details" and "Add to Cart (Price ₪39.50)" buttons. 4. "Hotel Transylvania 4" (Cineplex, Comedy, Jul 05 - Jul 12, Available) with "Show Details" and "Add to Cart (Price ₪39.50)" buttons. Each card includes a movie poster, title, cinema information, category, start and end dates, and status.

**7.6. ábra.** Bejelentkezés után

Ha a **7.6** ábrán vagyunk, akkor bármelyik "Add To Cart" gomb a **7.7** ábrához vezet.

eTickets  Movies

Search for a movie...

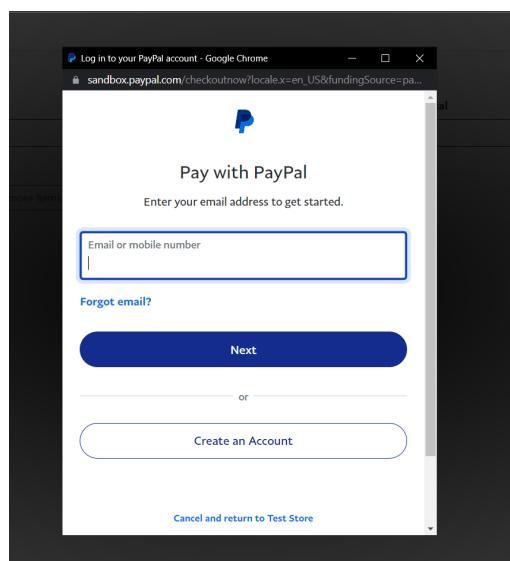
**You shopping cart summary**

Selected amount	Movie	Price	Subtotal
1	The Maze Runner	¤29.50	¤29.50  
		<b>Total:</b>	¤29.50

 [Buy now. Pay later.](#) [Add more items](#)

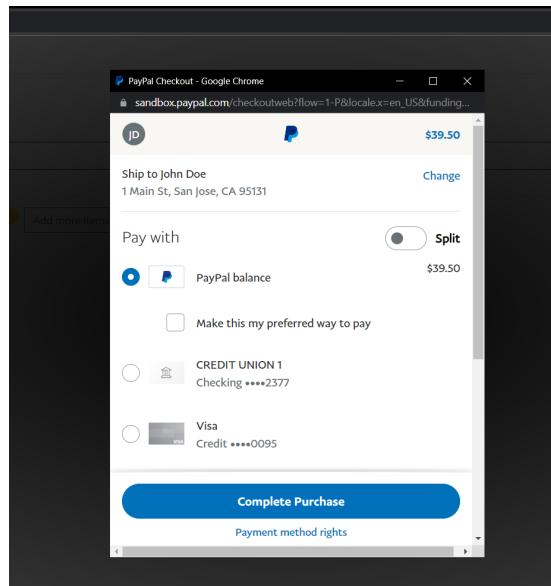
### 7.7. ábra. Kosár

Ha a 7.7 ábrán vagyunk és rákattintunk az "Add more items" gombra, akkor a 7.6 oldalt, lássuk. Ha pedig a "PayPal Checkout" sárga színű gombra kattintunk, akkor a 7.8 oldalt lássuk.



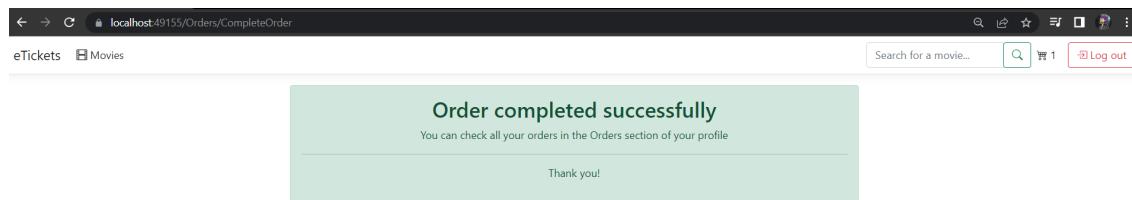
### 7.8. ábra. Pay Pal

Ha a 7.8 ábrán lévő oszlopokat kitöltsük egy megfelelő sandbox-os tesztelő konttal, akkor a 7.9 képet lássuk.



**7.9. ábra.** Vásárlás

A 7.8 ábrán látható "Complete Purchase" gomb pedig a 7.10 képhez vezet.



**7.10. ábra.** A rendelés sikeresen befejeződött

A "Log Out" gomb a 7.1 ábrához vezet.

## 7.0.2. Admin

Ha nem szimpla felhasználóként lépünk be a weboldalra, hanem admin-ként, akkor a 7.11 oldalt lássuk. Ezen az oldalon amik találhatóak voltak az előbbi oldalakon, azok pont úgy működnek itt is.

eTickets Movies Management

Search for a movie... Log out

<b>The Maze Runner</b>	<b>Parasites</b>	<b>The Walk</b>
This is the Maze Runner description	description	description
Cinema: Cinema OneE	Cinema: Cinema City	Cinema: Cineplex
Category: science_fiction	Category: Horror	Category: Drama
Start Date: 30 Jun 23	Start Date: 24 Jun 23	Start Date: 14 Jun 23
End Date: 03 Jul 23	End Date: 01 Jul 23	End Date: 04 Jul 23
Status:	Status:	Status:
<a href="#">Show Details</a>	<a href="#">Show Details</a>	<a href="#">Show Details</a>
<a href="#">Add to Cart (Price: \$29.50)</a>	<a href="#">Add to Cart (Price: \$39.50)</a>	<a href="#">Add to Cart (Price: \$39.50)</a>

**Hotel Transylvania 4**

This is Hotel Transylvania 4 movie  
description

Cinema: Cineplex

### 7.11. ábra. Bejelentkezés adminisztrátorként

Jobb alsó sarokban található "Add New" gomb a 7.12 ábrához vezet. Ezen az oldalon megadhassuk egy új film adatait és hozzáadhassuk az adatbázisunkhoz.

A "Log Out" gomb mellett látható a "Users" (egy emberke formájú logo mellett) logo, amire ha rákattintunk, akkor a 7.13 ábrát lássuk.

Ha valamelyik filmkártyán lévő ecset logóra kattintunk (ami pontosabban a film címe mellett látható), akkor a 7.14 ábrát lássuk. Ezen az ábrán módosítani lehet a film adatait.

Bal felső sarokban található logók melletti "Management" menügomb kattintására a 7.15 ábrán látható "gombokat" jeleníti meg.

### Create a new movie

Movie name	Select cinema
<input type="text"/>	<input type="text" value="Cinema City"/>
Start Date	Select a category
<input type="text" value="mm/dd/yyyy"/>	<input type="text" value="Action"/>
End Date	Select producer
<input type="text" value="mm/dd/yyyy"/>	<input type="text" value="Alice Dewey"/>
Price in \$	Select actor(s)
<input type="text"/>	<input type="text" value="Dylan O'Brien"/> <input type="text" value="Joseph Gordon-Levitt"/> <input type="text" value="Kim Min-gue"/> <input type="text" value="Selena Gomez"/>
Movie poster URL	
<input type="text"/>	
Movie description	
<input type="button" value="Create"/> <input type="button" value="Show all"/>	

### 7.12. ábra. Film hozzáadás

A következő képen látható az összes felhasználó, akik már regisztráltak a weboldalra:

Full Name	UserName	Email
Broccoli Mango	mango@gmail.com	mango@gmail.com
Applicaion User	app-user	user@etickets.com
Admin User	admin	admin@etickets.com
Giovanni Giorgio	giorgi@etickets.com	giorgi@etickets.com

**7.13. ábra.** Felhasználók

A 7.14 ábrán látható egy form, amit ha kitölt az adminisztrátor, akkor egy filmet fog hozzáadni az adatbázishoz illetve a főoldalhoz:

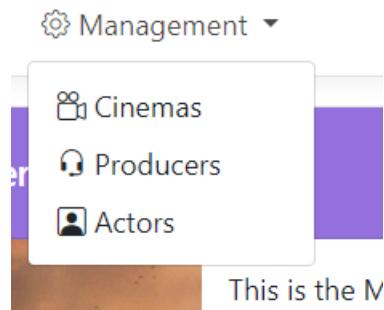
**Update movie**

Movie name	Select cinema
The Maze Runner	Cinema OneE
Start Date	Select a category
06/30/2023	science_fiction
End Date	Select producer
07/03/2023	Bong Joon-hooo
Price in \$	Select actor(s)
29.5	Dylan O'Brien Joseph Gordon-Levitt Kim Min-gue Selena Gomez
Movie poster URL	
<a href="https://w0.peakpx.com/wallpaper/817/577/HC">https://w0.peakpx.com/wallpaper/817/577/HC</a>	
Movie description	
This is the Maze Runner description	



**Update** **Show all**

**7.14. ábra.** Film frissítése



**7.15. ábra.** Management menu

A 7.15 látható gombok közül a "Cinemas" gomb a 7.16 képhez vezet, amin három gomb található mindegyik filmhez: "Edit", "Details", "Delete". Az "Edit" gomb a 7.17 ábrához vezet. Ez a film adatait módosítsa. A "Details" gomb a 7.18 ábrához vezet. Ez a film leírását jeleníti meg. A "Delete" gomb pedig kitörli a mozitermet az adatbázisból és a hozzá tartozó adatokat.

Logo	Name	Description	Actions
	Cinema OneE	Cinema One Brașov este destinația ideală pentru cinefilii care doresc să își petreacă timpul liber într-o atmosferă deosebită. Aici pot fi vizionate cele mai noi filme, într-un spațiu creat special pentru pasionații de film și spectacol.	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>
	Cinema City	Cinema City este cel mai mare operator de cinema din România și face parte din Cineworld Group, al doilea mare lanț de cinema din Europa.	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>
	Cineplex	Cineplex Targu Mures este cel mai nou multiplex din oraș, dotat cu cele mai avansate tehnologii de imagine și sunet din lume: Dolby Atmos, RealD Silver Screen și proiectoare cu laser.	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>

**7.16. ábra.** Mozik

## Update Cinema

Cinema Logo

The logo consists of the word "CINEMA" in a blue sans-serif font above the letters "one" in a larger, bold, blue font. A small blue star is positioned to the left of the "o".

Cinema Name

Cinema OneE

Cinema Description

Cinema One Brașov este destinația ideală pentru cinefilii care doresc să își petreacă timpul liber într-o  
atmosferă relaxată și confortabilă.

UpdateShow All

**7.17. ábra.** Mozi frissítése

## Details for Cinema OneE

Cinema Logo

The logo consists of the word "CINEMA" in a blue sans-serif font above the letters "one" in a larger, bold, blue font. A small blue star is positioned to the left of the "o".

Cinema Name

Cinema OneE

Cinema Description

Cinema One Brașov este destinația ideală pentru cinefilii care doresc să își petreacă timpul liber într-o  
atmosferă relaxată și confortabilă.

EditShow All

**7.18. ábra.** Mozi részlete

# Ábrák jegyzéke

3.1. MVC . . . . .	12
3.2. DBeaver . . . . .	13
4.1. Cineplexx . . . . .	15
4.2. Cinema One . . . . .	16
4.3. Cinema City . . . . .	16
5.1. Use Case Diagram . . . . .	18
6.1. Front-end arhitektúra . . . . .	21
6.2. Rendszer arhitektúra . . . . .	27
6.3. DbContext . . . . .	27
6.4. Adatbázis . . . . .	28
6.5. Docker Desktop . . . . .	29
7.1. Főoldal/Filmek . . . . .	31
7.2. Részletek . . . . .	32
7.3. Bejelentkezés . . . . .	32
7.4. Regisztrálás . . . . .	32
7.5. Sikeresen létrehozva . . . . .	33
7.6. Bejelentkezés után . . . . .	33
7.7. Kosár . . . . .	34
7.8. Pay Pal . . . . .	34
7.9. Vásárlás . . . . .	35
7.10. A rendelés sikeresen befejeződött . . . . .	35
7.11. Bejelentkezés adminisztrátorként . . . . .	36
7.12. Film hozzáadás . . . . .	36
7.13. Felhasználók . . . . .	37
7.14. Film frissítése . . . . .	37
7.15. Management menu . . . . .	38
7.16. Mozik . . . . .	38
7.17. Mozi frissítése . . . . .	39
7.18. Mozi részlete . . . . .	39

# Összefoglaló

Dolgozatom egy "eTickets" című webalkalmazás készítéséről szolt, amely lehetőséget nyújt a felhasználóknak a könnyű filmjegyek onlinne vásárlására.

A webalkalmazásomat különböző programozási technológiák segítségével valósítottam meg. Először ismertettem a programozási technológiák elméleti alapjait, kutattam a web áruházak tervezésében, majd összehasonlítottam az általam létrehozott webalkalmazást a már működő romániai webalkalmazásokkal. Ez után kiemeltem az alkalmazásom egyediségét és a konkurencia előnyeit.

Ezt követően részletesen megvizsgáltam a szoftvereket, amelyek segítségével megvalósítottam a webalkalmazásomat. Az ASP.NET keretrendszeret használtam a webalkalmazás fejlesztéséhez, valamint Docker-t használtam a környezet függetlenségének biztosításához.

Jövőbeli terveim között szerepel a sikeres és valós jegyvásárlási folyamatok továbbfejlesztése. Szeretném, ha a webalkalmazásom még kényelmesebb és szórakoztatóbb élményt nyújtana a filmszerető felhasználók számára. Ez magába foglalhatja az új funkciók bevezetését, interaktív elemek hozzáadását vagy akár a felhasználói felület továbbfejlesztését.

Célom, hogy az "eTickets" webalkalmazás kiváló minőségű szolgáltatást nyújtsan és pozitív hatást kínáljon az emberek minden nap szórakoztatására, amit a filmnézésből szerez, miközben könnyebbé teszi a filmjegyek online vásárlását.

A dolgozatom a következő GitHub link-ről töölhető le: [eTickets](#)

# Irodalomjegyzék

- [1] .NET Framework, I. (2017).
- [2] .NET Framework. Tendulkar, S. S. (2022).
- [3] ASP.NET MVC. [learn.microsoft.com](https://learn.microsoft.com/en-us/aspnet/mvc).
- [4] Automated Dockerization of Web Applications Developed in Various Languages. Doctoral dissertation, Dublin, National College of Ireland.
- [5] Kurzusok. [udemy](https://www.udemy.com/).
- [6] C#: The Complete Reference (Osborne Complete Reference Series).
- [7] [paypal.developer.sandbox](https://developer.paypal.com/test/sandbox).
- [8] Az ábrákat a Diagrams.net segítségével készítettem. [draw.io](https://draw.io).