



# MOVIE CATALOG SYSTEM

Работа по курсов проект

- Математическа гимназия „Академик Кирил Попов“ 4001 Пловдив, ул. „Чемшир“ 11  
тел.: + 359 32 643 157, E – mail: [omg@omg-bg.com](mailto:omg@omg-bg.com)

Ръководител:

- Росен Вълчев
- Изработен от: Айлин Шабан, Иван Мишевски

# **Съдържание**

## **1. Цели**

## **2. Разпределение на ролите**

## **3. Ниво на сложност на проекта**

### **3.1 Entity Framework**

### **3.2 Адаптивен и Модерен Потребителски**

## **Интерфейс**

### **3.3 Бизнес логика**

## **4. Инсталация и деинсталация**

## **5. Основни етапи в реализирането на проект**

### **5.1 Потребители**

## **6. Реализация**

### **6.1 Архитектура**

### **6.2 База данни**

### **6.3 Описание на приложението**

### **6.4 Използвани технологии**

### **6.5 Използвани езици за програмиране**

### **6.6 Описание на приложението**

## **7. Развитие и нововъведения**

## **8. Заключение**

## **9. Използвана литература**

### **9.1 Книги**

### **9.2 Линкове, сайтове**

# 1.Цели

Целта на проекта е да се разработи интуитивно и функционално уеб приложение, което улеснява потребителите при откриването и управлението на информация за филми. Приложението ще предлага удобен интерфейс за търсене, визуализация и оценка на филми по различни критерии, като осигурява бърз и лесен достъп до подробности за заглавията, жанровете, режисьорите и потребителските ревята.

Основните цели на проекта включват:

- Функционалност за търсене на филми: Приложението ще позволява на потребителите да търсят филми по заглавие, жанр или година. Това ще спомогне за лесното намиране на желаните филми без необходимост от продължителни търсения в различни платформи.
- Категоризиране и визуализация на филми: Филмите ще бъдат организирани в различни категории и жанрове, което ще даде на потребителите ясен преглед на наличните заглавия и ще улесни избора им.
- Оценки и ревята: Приложението ще предоставя възможност за разглеждане и оставяне на потребителски оценки и коментари за филми, като по този начин ще подпомага информирания избор на зрителите.

Тези функционалности ще подобрят потребителското изживяване, като осигурят удобен и ефективен начин за откриване и организиране на информация за филми.

## 2. Разпределение на ролите

Обработката на софтуера е разпределена между екипа, както следва:

Айлин Шабан: Отговорна за разработка на презентационния слой, потребителския интерфейс, базата данни и документация.

Иван Мишевски: Отговорен за функционалността, тестване, базата данни, документация и управление на версиите чрез Git.

## 3. Ниво на сложност на проекта

Определяме нивото на сложност на проекта като умерено.

Сблъскахме се с някои предизвикателства, които успешно преодоляхме:

- Entity Framework Едно от първите препятствия беше да се научим как да използваме Entity Framework за управление на базата данни. Сблъскахме се с проблеми със съвместимостта на версиите, но след проучване и тестване успяхме да интегрираме EF успешно.
- HTML за потребителски интерфейс Изборът на подходящ дизайн за потребителския интерфейс беше предизвикателство. Използвахме стандартни HTML елементи и CSS за създаване на прост, но функционален интерфейс. Вместо да използваме готови библиотеки като Bootstrap, решихме да изградим интерфейса ръчно, за да имаме пълен контрол върху дизайна и функционалността.

## 3.1 Entity Framework

Едно от първите препятствия, които трябваше да преодолеем, беше да се научим как да използваме Entity Framework. Сблъскахме се с проблеми с инсталацията и проблеми със съвместимостта на версиите, което добави сложност и изискваше допълнително отстраняване на неизправности. Въпреки това, след известно проучване и постоянство, успяхме да стартираме Entity Framework.

## 3.2 Адаптивен и Модерен Потребителски Интерфейс

За да осигурим удобство и интуитивност при използването на уеб приложението, дизайнът се базира на принципите на адаптивния (responsive) и модерен потребителски интерфейс (UI/UX). Това включва:

- **Отзивчив дизайн:** Интерфейсът ще се адаптира динамично към различни устройства и резолюции (настолни компютри, таблети, мобилни устройства), осигурявайки оптимално визуално и функционално изживяване.
- **Модулна структура:** Използване на модулен подход за проектиране, който улеснява бъдещи разширения и поддръжка на нови функционалности.

## 3.3 Бизнес логика

За каталог за филми бизнес логиката може да включва следните елементи:

- **Обработка на заявки за търсене и филтриране:** Определяне на алгоритми, които обработват потребителските заявки за търсене на филми по заглавие, жанр, година и други критерии. Това включва динамично филтриране, сортиране и оптимизиране на резултатите, така че потребителите да намират желаната информация бързо и ефективно.
- **Управление на ревюта и оценки:** Логиката за добавяне, валидиране и агрегация на потребителски ревюта и оценки. Това включва изчисляване на среден рейтинг за всеки филм, модериране на съдържанието и предотвратяване на фалшиви или неподходящи коментари.
- **Потребителско управление и автентикация:** Обработка на процесите по регистрация, вход и управление на потребителските профили, което гарантира, че само упълномощени потребители могат да добавят ревюта, рейтинги или да извършват други специфични действия. Тук се определят и нивата на достъп според ролите на потребителите.
- **Валидация и обработка на данни:** Осигуряване, че всички входни данни (като данни за нови филми, ревюта, рейтинги и т.н.) отговарят на предварително дефинираните правила и формати, за да се запази целостта на информацията в системата.
- **Интеграция с външни системи:** При необходимост, бизнес логиката може да включва интеграция с външни API-та или бази данни за извличане на допълнителна информация за филмите (например трейлъри, актьорски състав, технически спецификации и т.н.) и автоматично обновяване на данните в каталога.

## 4. Инсталация и деинсталация

Преди да бъде използван разработения софтуер е необходимо да се премина през последователност от стъпки за неговата инсталация и настройки за експлоатация.

Нека започваме:

- Инсталирайте Visual Studio 2022
- Уверете се, че .NET 9 е инсталиран
- Инсталирайте Microsoft SQL Server
- Инсталирайте SQL Management Studio

1.Клонирайте кода, като използвате следната команда: `git clone`

2.Отворете проекта във Visual Studio. С помощта на SQL Management Studio създайте базата данни. Отидете на Visual Studio

3.Отидете на `appsettings.json` и променете низовете за връзка (connection strings), за да сочат към вашата база данни

4.Отворете Package Manager Console

5.Въведете следната команда, за да приложите миграциите и да създадете базата данни: `Update-Database`

6. Стартирайте приложението чрез Visual Studio (Ctrl + F5) или чрез командата: `dotnet run`

7.Приложението ще се стартира на локален сървър (по подразбиране на <http://localhost:5000/>).

8.За деинсталация, изтрийте папката, в която се намира проекта и изтрийте и локалната база от данни.



## 5. Основни етапи при създаването на проекта

### 5.1 Потребители

Приложението поддържа два вида потребители:

- Администратори: Могат да извършват CRUD операции върху всички данни.
- Обикновени потребители: Могат да преглеждат данните, но не могат да ги редактират или изтриват.

### 5.1 Модул регистрирани потребители

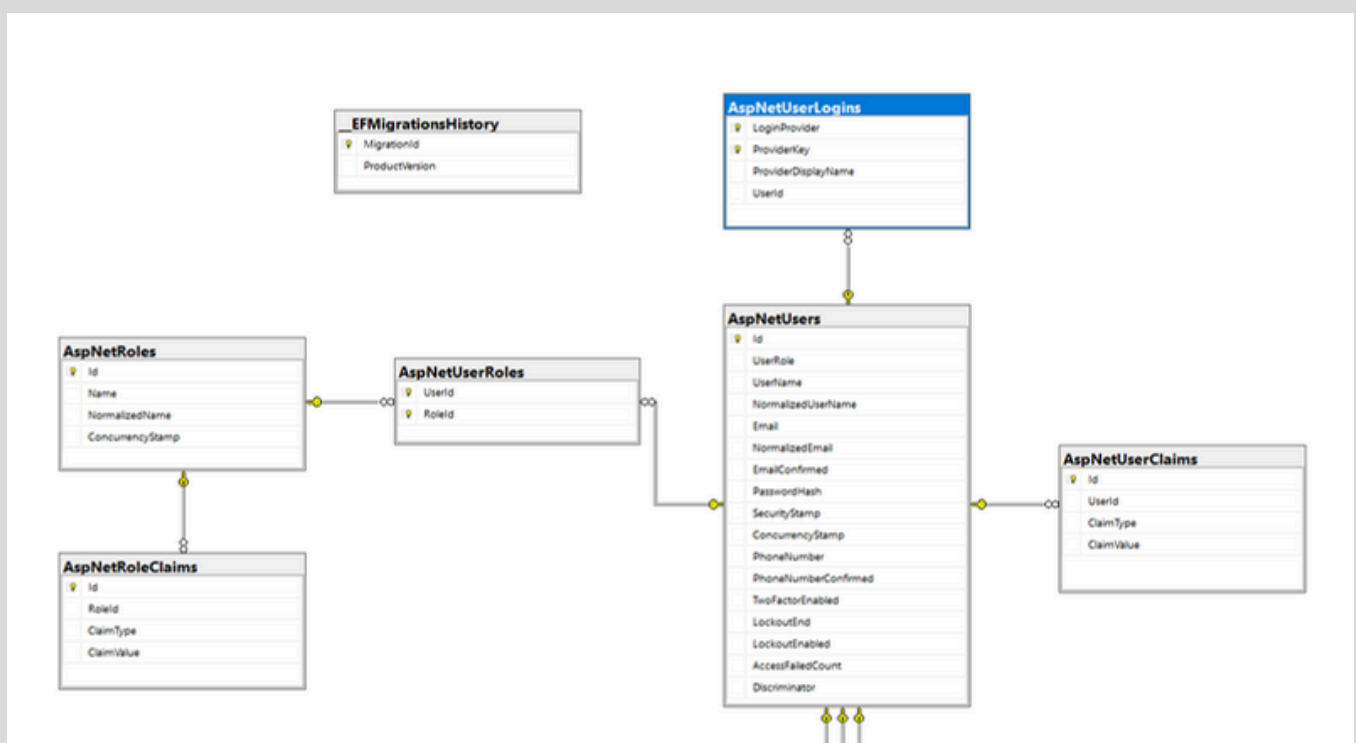
Този модул позволява на всички потребители да се регистрират и да създадат личен профил, чрез който да се идентифицират и използват всички предоставени функционалности на уеб приложението.

## 6. Реализация

Софтуерът е изграден от един проект – MovieCatalog.

### 6.1 Архитектура

### 6.2 Бази данни

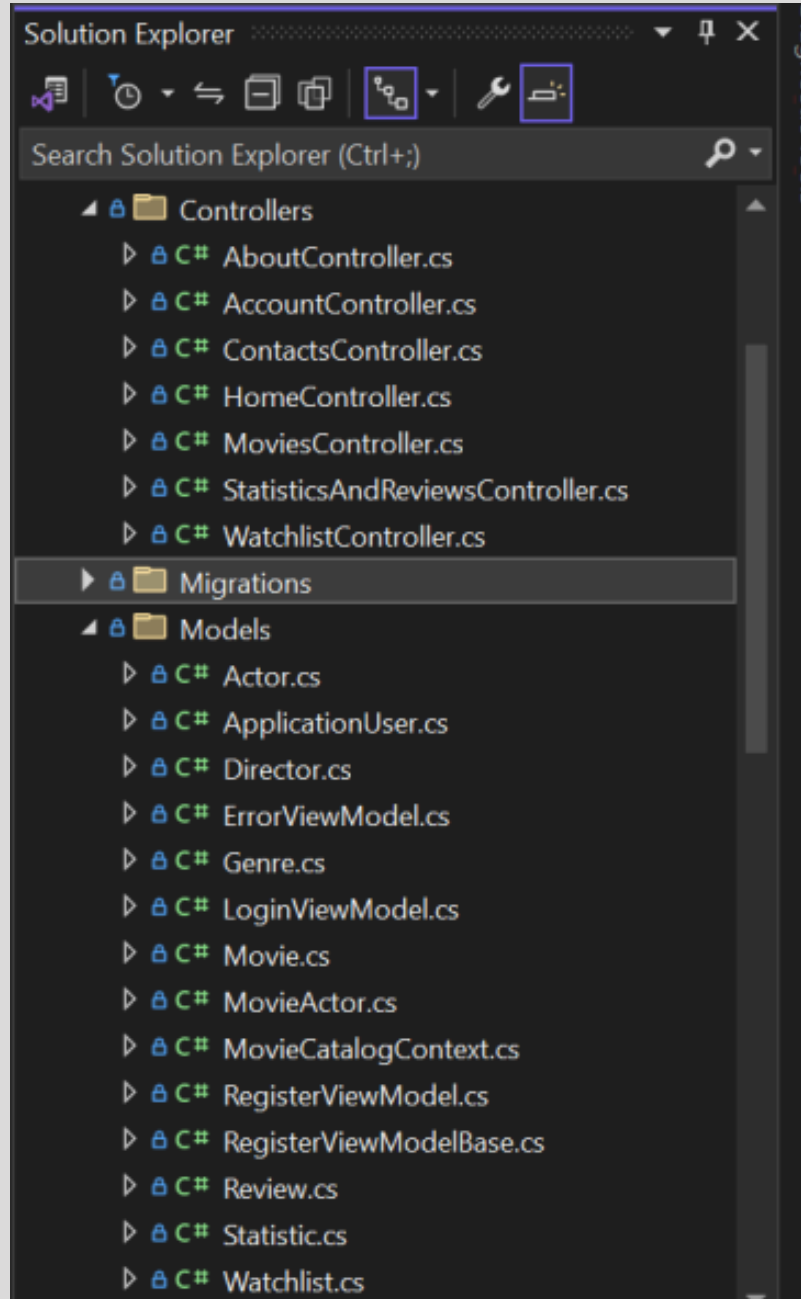






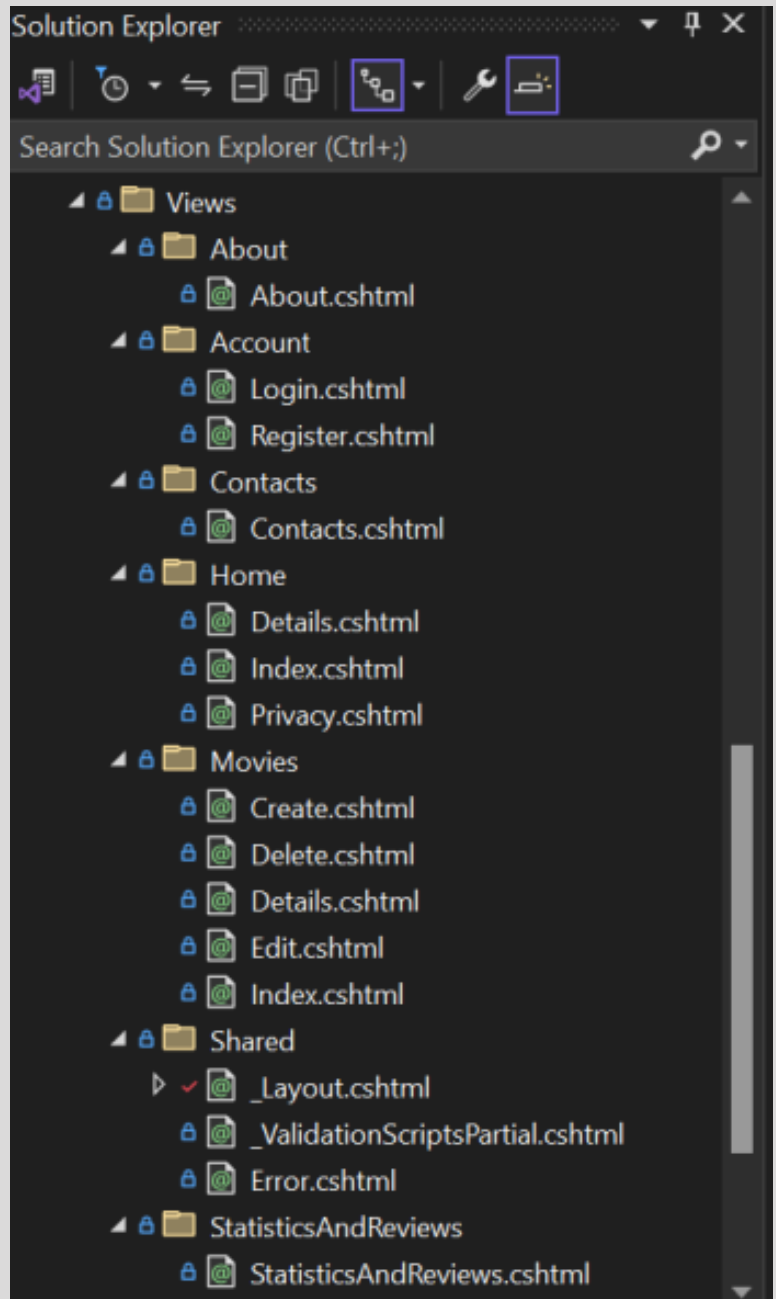
## 6.3 MovieCatalog

- **Controllers** – тук се съдържа бизнес логиката. Реализира се връзка с базата данни и модифициране на информация – добавяне, редактиране и други сложни функционалности
- **Migrations** – папка, генерирана от EntityFramework, която служи за съхраняване на миграциите.
- **Models** – основните модели, които представят моделите в базата от данни.
- **Views** – тук се съдържат интерфейсите на приложението



## 6.3 MovieCatalog

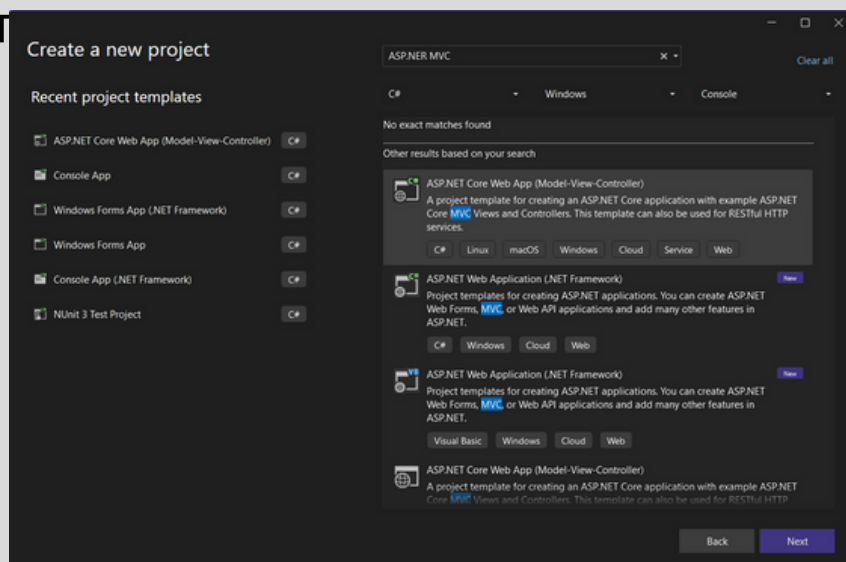
- Controllers – тук се съдържа бизнес логиката. Реализира се връзка с базата данни и модифициране на информация – добавяне, редактиране и други сложни функционалности
- Migrations – папка, генерирана от EntityFramework, която служи за съхраняване на миграциите.
- Models – основните модели, които представят моделите в базата от данни.
- Views – тук се съдържат интерфейсите на приложението



## 6.4 Използвани технологии

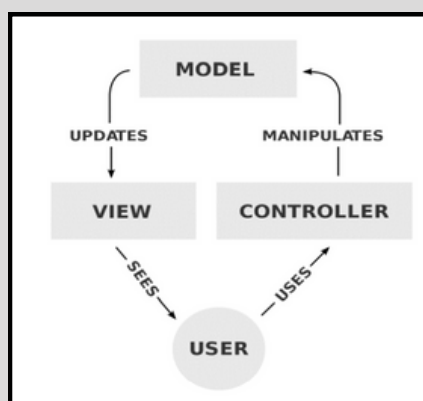
За разработката на уеб приложението се използват следните технологии:

- **.NET Framework 9:** Осигурява стабилна и съвременна платформа за разработка на приложението, като поддържа широк набор от функционалности и инструменти за създаване на мащабируем софтуер.
- **ASP.NET MVC:** Използва се за създаване на динамичен и добре структуриран уеб интерфейс, като разделя логиката на приложението на три основни компонента – Model, View и Controller. Това подобрява поддръжката и развитието на кода.
- **Entity Framework:** Този ORM (Object-Relational Mapping) инструмент улеснява работата с базата данни чрез обектно-ориентиран подход, което намалява сложността при управлението на данните и подобрява производителността.
- **HTML/CSS:** Използват се за разработка и стилизиране на потребителския интерфейс, като гарантират отзивчив дизайн и приятно визуално изживяване за крайните потребители.
- **Git:** Системата за контрол на версиите улеснява управлението на изходния код, позволявайки съвместна работа, проследяване на промените и лесно интегриране на нови функционалности.



- **Visual Studio 2022:** Visual Studio 2022 е мощна интегрирана среда за разработка (на английски: integrated development environment, IDE) на софтуерни приложения за Windows и за платформата .NET Framework. Използва се за разработка на конзолни и графични потребителски интерфейси приложения, както и Windows Forms или WPF приложения, уеб сайтове, уеб приложения и уеб услуги на всички поддържани платформи от Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight. Visual Studio предоставя мощна интегрирана среда за писане на код, компилиране, изпълнение, дебъгване (както за високо така и за машинно ниво), тестване на приложения, дизайн на потребителски интерфейс (форми, диалози, уеб страници, визуални контроли и други), моделиране на данни, моделиране на класове, изпълнение на тестове, пакетиране на приложения и стотици други функции
- **Microsoft SQL Server:** Microsoft SQL Server е сървърна система за управление на бази от данни (и по-точно на релационни бази от данни) на компанията Microsoft. Microsoft SQL Server е предназначена за управление на големи сървърно базирани БД, за разлика от MS Access, която е desktop базирана и не е предназначена за управление на големи корпоративни БД.
- **Microsoft.VisualStudio.TestTools.UnitTesting** - е тестов framework, който е част от Visual Studio IDE и се използва за автоматизирано модулно тестване на .NET Framework приложения. Той позволява на разработчиците да пишат и изпълняват тестове във Visual Studio и предоставя набор от функции и инструменти за управление и анализ на резултатите от тестовете.

- Архитектурният модел Model-View-Controller (MVC) разделя уеб приложението на три основни компонента, всеки от които има ясно дефинирана отговорност:
- Модели (Models): Те представляват бизнес логиката и данните на приложението. В контекста на нашия каталог за филми, моделите съдържат класове, които управляват информация за филми, ревюта, потребители, жанрове и други. Моделите са отговорни за комуникацията с базата данни чрез ORM (например Entity Framework) и изпълняват CRUD операции, които гарантират целостта на данните.
- Изгледи (Views): Изгледите отговарят за визуалното представяне на данните. Те използват HTML, CSS и шаблони (например Razor шаблони в ASP.NET MVC), за да генерират потребителски интерфейс, който е интуитивен и лесен за навигиране. Изгледите получават данните от контролерите и ги показват по начин, който улеснява потребителя в търсенето и разглеждането на филмите в каталога.
- Контролери (Controllers): Контролерите действат като посредници между потребителските заявки, моделите и изгледите. Когато потребител направи заявка (например търсене на филм, филтриране по жанр или добавяне на ревю), контролерът я получава и изпълнява необходимите операции чрез комуникация с моделите. След обработката, контролерът избира подходящ изглед и му предоставя всички необходими данни за визуализацията на резултатите.





## 6.5 Използвани езици за програмиране

**1) C#** - C# е обектно-ориентиран език за програмиране, разработен от Microsoft, като част от софтуерната платформа .NET. Стремехът още при създаването на C# езика е бил да се създаде един прост, модерен, обектно-ориентиран език с общо предназначение. Основа за C# са C++, Java и донякъде езици като Delphi, VB.NET и C. Той е проектиран да балансира мощност (C++) с възможност за бързо разработване (Visual Basic и Java). Те представляват съвкупност от дефиниции на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява.

**2) T-SQL (Transact-SQL)** - е диалект на SQL (Structured Query Language), използван от SQL Server на Microsoft и Azure SQL база данни. T-SQL включва всички стандартни SQL функции, като заявки и манипулиране на данни в релационни бази данни, но включва и допълнителни функции, специфични за SQL Server. Някои от допълнителните функции, предлагани от TSQL, включват съхранени процедури, тригери, дефинирани от потребителя функции и отчети за контрол на транзакциите. Тези функции помагат на разработчиците да пишат сложни и ефективни приложения за бази данни и им позволяват да се възползват напълно от възможностите на SQL Server.

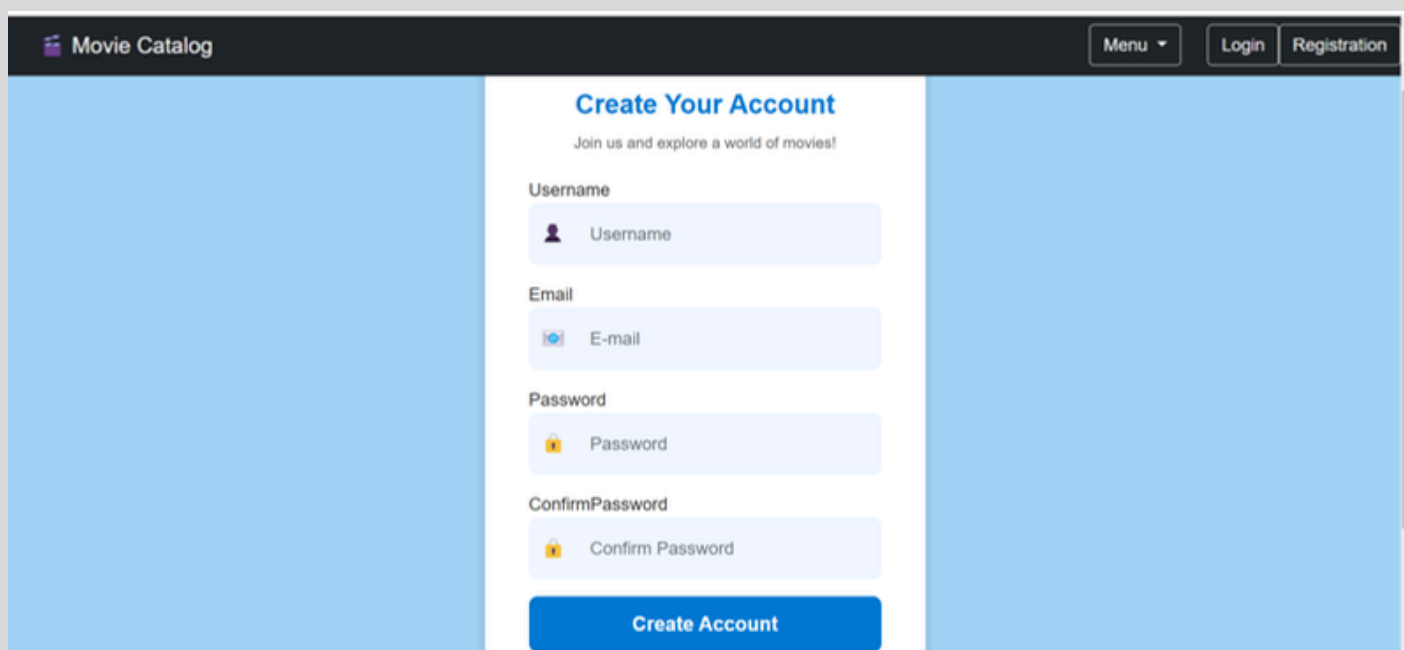
**3) HTML/CSS** - Използват се за изграждане и стилизиране на потребителския интерфейс, като осигуряват структурирано и визуално привлекателно представяне на съдържанието.



## 6.6 Описание на приложението

### 6.6.1 Register

Register.cshtml: Форма за регистрация на нови потребители, където се въвежда основна информация (потребителско име, имейл, парола и т.н.).



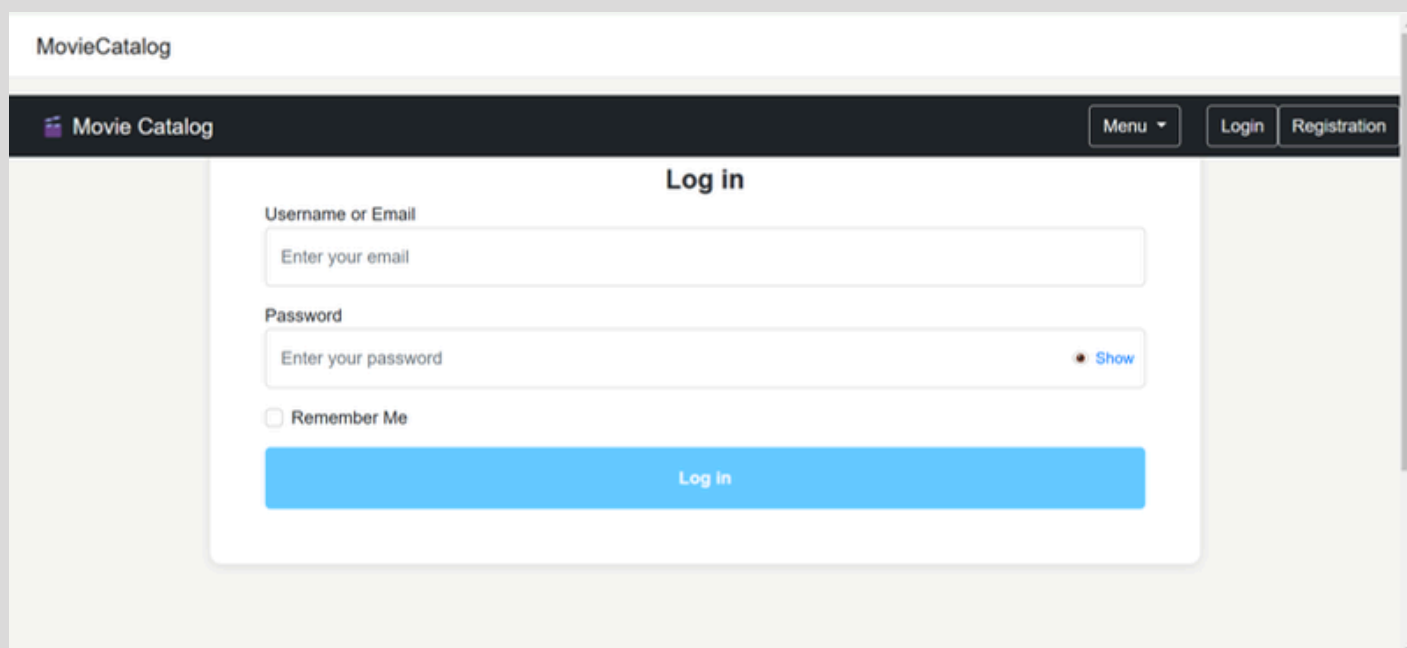
The screenshot shows a web application titled "Movie Catalog" with a dark header. In the top right corner of the header are links for "Menu", "Login", and "Registration". The main content area has a light blue background. Centered on this background is a white box titled "Create Your Account" with the subtitle "Join us and explore a world of movies!". Inside this box are four input fields: "Username" (with a person icon), "Email" (with an envelope icon), "Password" (with a lock icon), and "Confirm Password" (with a lock icon). Below these fields is a blue button labeled "Create Account".

Ако регистрацията е успешна, потребителят може да бъде автоматично вкаран в системата (login) или пренасочен към страницата за вход, в зависимост от настройките на приложението. При неуспешна регистрация (например дублиращо се потребителско име или несъответствие при паролата) се връща съобщение за грешка и формата се показва отново за корекция на данните.

### 6.6.2 LoginForm

Login.cshtml: Форма за вход в системата, където потребителите въвеждат своите идентификационни данни.

“LoginForm” е форма, използвана за влизане в системата. Тя включва текстови полета за въвеждане на имейл и парола на потребителя. Полето за парола е настроено така, че да приема за валидни само налични в базата потребители. Тази настройка гарантира, че паролата остава скрита и защитена от всякакви неоторизирани зрители. Въпреки това, за да се даде възможност на потребителите да видят въведената от тях парола, се включва и иконката за показване на парола под формата на изображение на око. Като се щракне върху иконата, потребителят може да превключи видимостта на въведената от тях парола, което му позволява да види текста, който е въвел, вместо само точките. Тази функция е полезна в случай, че потребителят е забравил паролата си или я е написал неправилно, тъй като може визуално да потвърди текста, който е въвел, преди да щракне върху бутона за влизане. Щракването върху бутона „Login“ потвърждава информацията на потребителя.



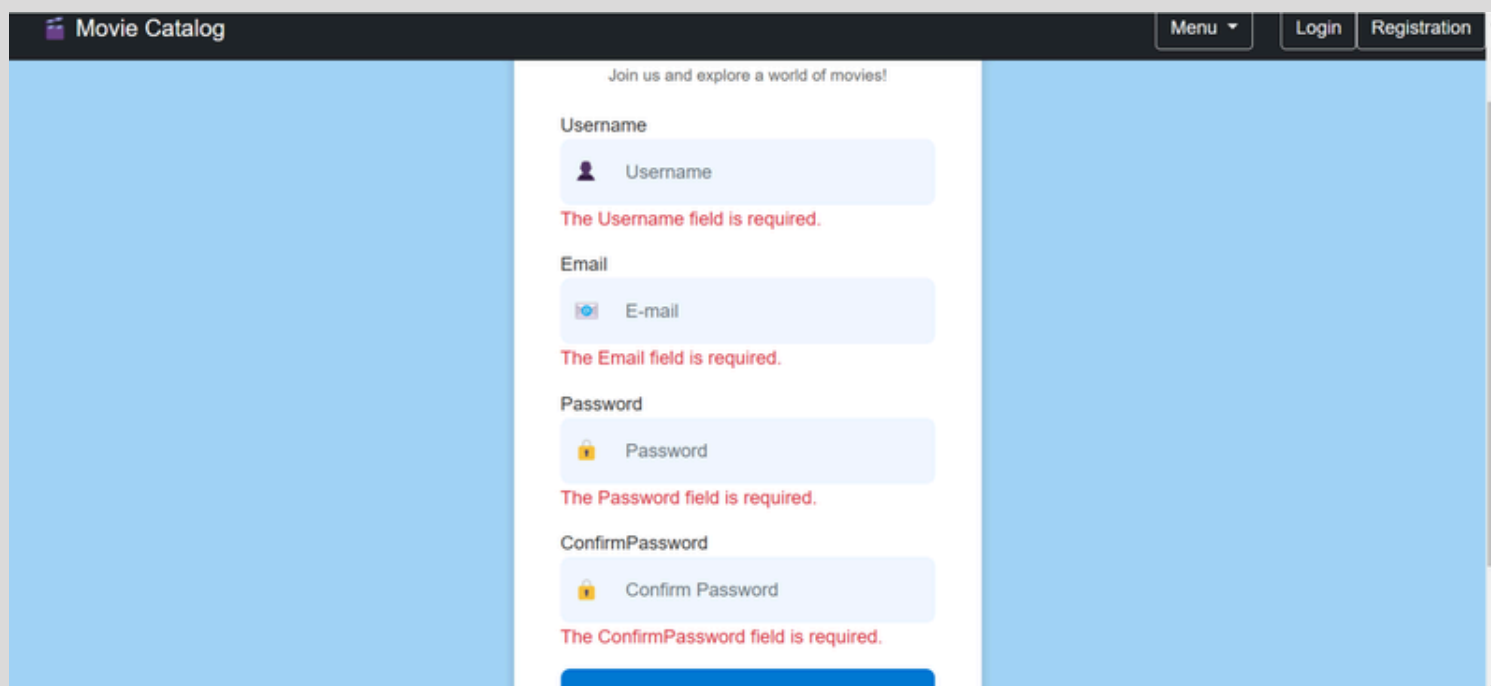
The screenshot displays the 'MovieCatalog' login interface. At the top, a dark navigation bar contains the 'Movie Catalog' logo, a 'Menu' dropdown, and 'Login' and 'Registration' buttons. The main content area features a 'Log in' form with the following elements: a 'Username or Email' field with placeholder text 'Enter your email'; a 'Password' field with placeholder text 'Enter your password' and a 'Show' toggle icon; a 'Remember Me' checkbox; and a prominent blue 'Log in' button.

Регистрация и Вход: Потребителите могат да създават акаунт и да влизат в системата. След успешен вход имат достъп до допълнителни функции, като добавяне/редактиране на филми, писане на ревьюта и създаване на личен списък за гледане (Watchlist).

## Тест за опит за влизане с невалидни стойности

При опит за изпращане на формата без да са попълнени коректно полетата, потребителят няма да има възможност да се регистрира. Всички полета са задължителни като:

- Паролата трябва да съдържа 8 или повече символа (главни и малки букви, цифри)



The screenshot shows a web application titled "Movie Catalog" with a navigation bar containing "Menu", "Login", and "Registration" links. The main content area features a registration form with the heading "Join us and explore a world of movies!". The form includes four input fields: "Username", "Email", "Password", and "ConfirmPassword". Each field is accompanied by a red error message: "The Username field is required.", "The Email field is required.", "The Password field is required.", and "The ConfirmPassword field is required." respectively. The form is set against a light blue background.

### 6.6.3 Home Page

След успешно регистриране потребителят има достъп до "Home page" на приложението, който съдържа топ три филма от базата, меню и търсачка имаща филтър за търсене по заглавие, жанр, година. Бутоните "Details" водят до страницата, в която потребителят има достъп да ви информацията за филмите и има правото да добавя ревя и да оценява филма (по скала от 1 до 10), след което информацията се изпраща в базата и се актуализира за дадения филм.

## Welcome to the Movie Catalog!

### Top movies



#### Laugh Out Loud

A series of hilarious misadventures unfolds when an awkward office worker tries to impress his coworkers during a company retreat.

Genre: Comedy

[Details](#)



#### Nightfall

A small town is haunted by mysterious disappearances, and a group of friends must uncover the dark truth before it's too late.

Genre: Horror

[Details](#)



#### Quest for Glory

A young adventurer embarks on a perilous journey to reclaim a lost artifact, facing mythical creatures and ancient secrets.

Genre: Adventure

[Details](#)

## 6.6.4 Movies

Когато изберем от падащото меню страницата Movies навигацията ни води до страница, в която виждаме всички филми. Страницата има подобна структура на home page. Бутонът “Details” същата функционалност. И когато се натисне ни води до следната страницата:



### Laugh Out Loud

Genre: Comedy

Director: Taika Waititi

Movie Actors: Emma Stone, Daniel Kaluuya

Description: A series of hilarious misadventures unfolds when an awkward office worker tries to impress his coworkers during a company retreat.

Release Year: 2022

[Add to Watchlist](#)

[Back to Movies](#)

### Leave a Review

Write your review here...

Rating:

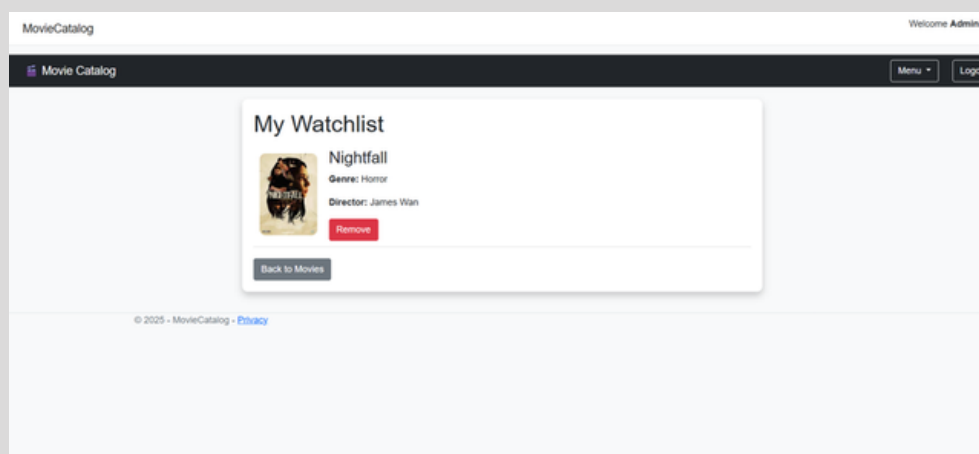
1

[Submit Review](#)

Потребителят, както може да добави ревю и рейтинг и да ги потвърди с бутона “Submit Review“, за да се актуализират в базата и да се покажат, така може да добави филма в своя лист за гледане чрез бутона “Add to Watchlist“. Когато се натисне бутонът, чрез функционалностите на страницата, потребителят ще вижда интерфейса на страницата Watchlist. Бутонът за “Back to Movies“ ни връща в страницата за всички книги.

## 6.6.5 Watchlist

Watchlist е функционалност в уеб приложението, която позволява на потребителите да създават и управляват личен списък с филми, които желаят да гледат по-късно. Тя работи в тясна връзка със Details страницата на всеки филм, където потребителят може да избере бутона Add to Watchlist, за да добави съответното заглавие в своя списък. Чрез бутона “Remove“ може потребителят да премахне филма от своя списък.

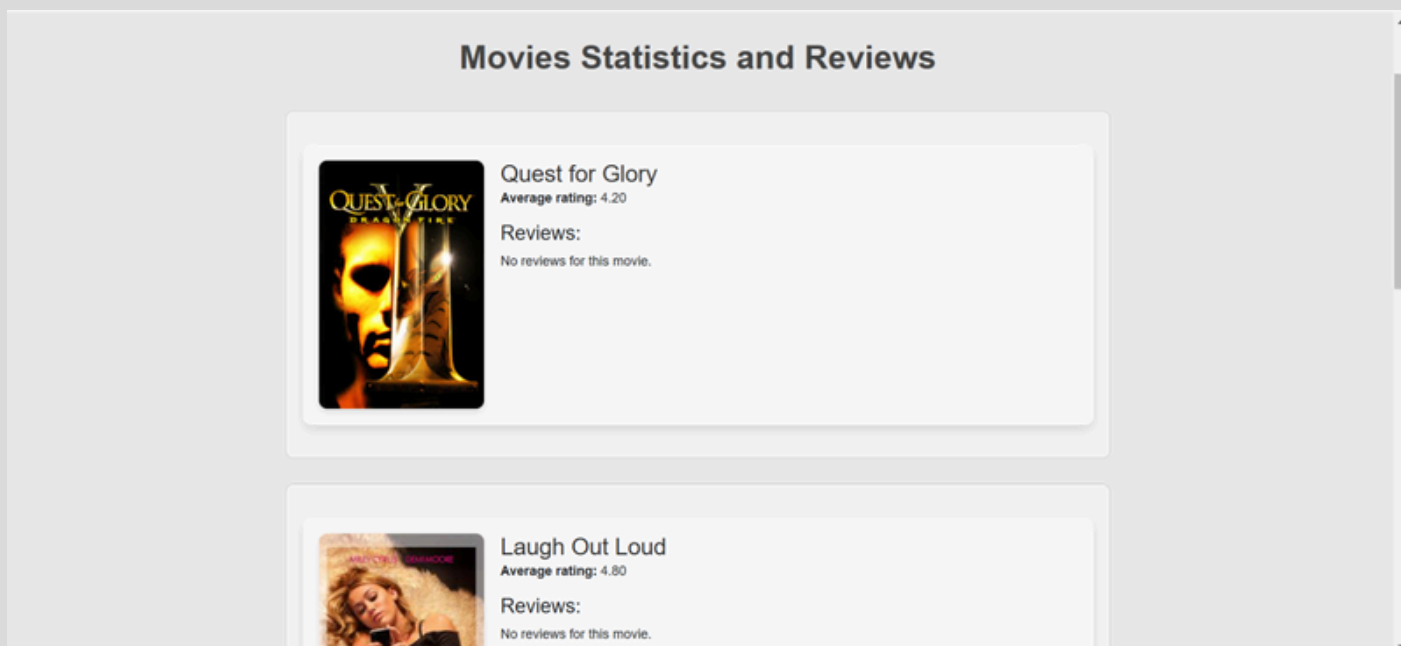


## 6.6.6 Statistics and Reviews

Страницата „Statistics and Reviews“ предоставя на потребителите детайлна информация за рейтинга и ревютата на всеки отделен филм. Тя служи като централизирано място, където се събират и визуализират мненията на регистрираните потребители, както и обобщена статистика за филма. Ето какви основни елементи и функционалности може да съдържа тази страница:

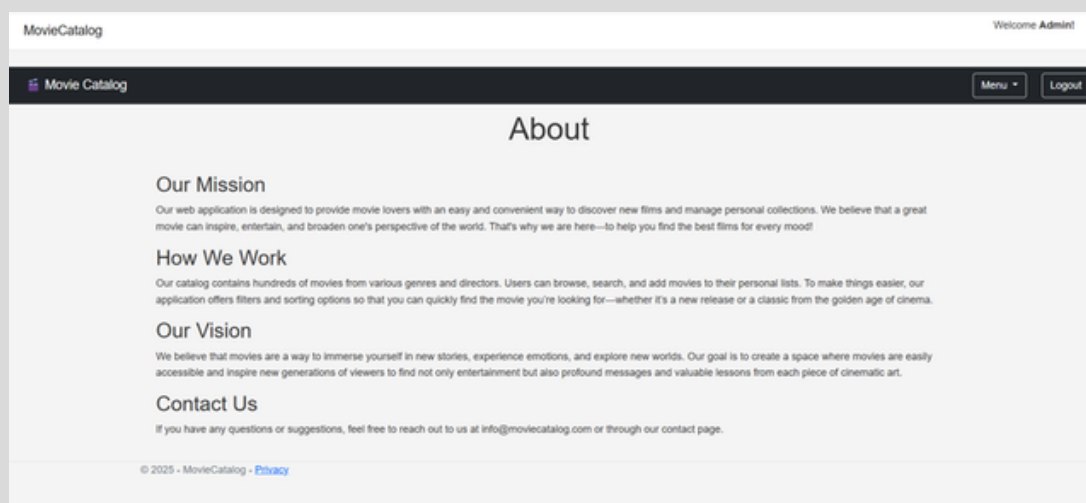
## 1) Обобщена статистика за филма

- Средна оценка (Average Rating): Изчислява се автоматично на база всички въведени от потребителите рейтинги (например от 1 до 5 звезди). Тази стойност дава бърз поглед върху цялостното впечатление от филма.
- Брой ревюта и оценки: Показва колко ревюта са налични. По този начин се вижда активността и популярността на заглавието.



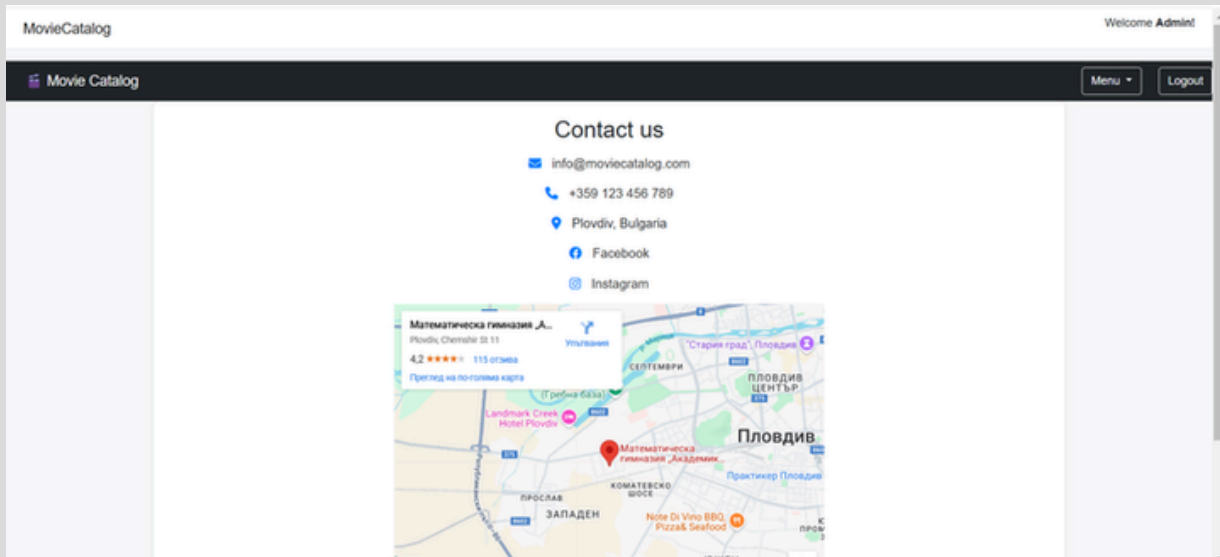
## 6.6.7 About us

Страницата About Us предоставя на потребителите детайлна информация за ценностите, начина на работа и визията на нашето уеб приложение, насочено към любителите на филми.



## 6.6.8 Contacts

Страницата Contact предоставя лесен и бърз начин за свързване с екипа на MovieCatalog. Тя съдържа ясна и изчистена информация, която позволява на потребителите да намерят всички необходими данни за контакт:



## 6.6.9 Admin account

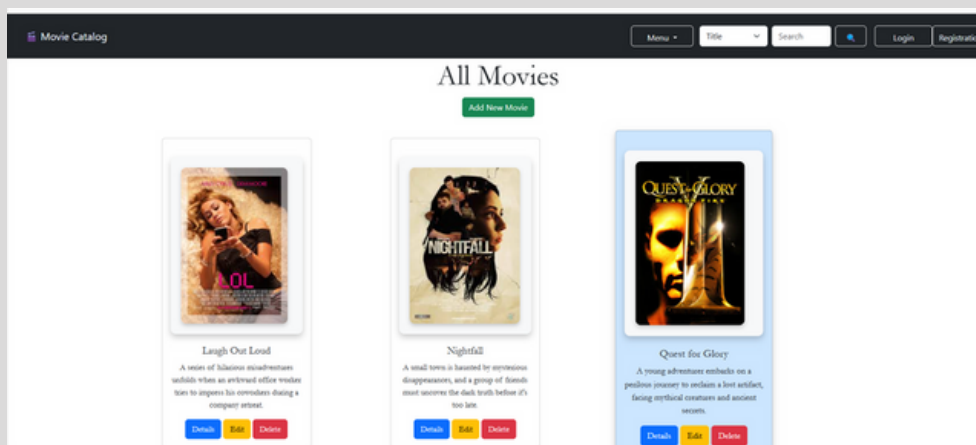
Администраторският акаунт е специализиран потребителски профил с разширени привилегии, предназначени за управление на съдържанието в каталога на филмите. Този акаунт има пълен контрол върху CRUD операциите (създаване, четене, редактиране и изтриване) на филмовите записи.

Основни характеристики и функционалности:

- **Добавяне на филми:** Чрез администраторския акаунт, чрез специален бутон "Add New Movie" в детайлите на каталога или в административния панел, администраторът може да отвори формуляр за въвеждане на нов филмов запис. Тук се попълват данни като заглавие, описание, жанр, режисьор, актьори, година на издаване, рейтинг и други релевантни атрибути.



## 6.6.9 Admin account



- **Редактиране на филми:** При преглеждане на детайлната страница на конкретен филм, администраторът вижда допълнителен бутон "Edit". Този бутон води до форма, в която може да се коригират или обновят съществуващите данни за филма. Това позволява бързо актуализиране на информацията според променящите се нужди или корекции.
- **Изтриване на филми:** Администраторът има правото да изтрива филмови записи, които вече не са валидни или желани. Допълнителният бутон "Delete" на детайлната страница активира процес за потвърждаване на изтриването, което гарантира, че записа няма да бъде премахнат по погрешка.
- **Допълнителни контролни механизми:** Всички тези действия се осъществяват чрез допълнителни бутони и специално проектирани интерфейсни елементи, които са видими само за потребители с администраторски права. Това осигурява сигурност и предотвратява нежелан достъп до административните функции от неупълномощени потребители.

Този акаунт е от съществено значение за поддържането и обновяването на каталога, като гарантира, че информацията за филмите е винаги актуална и точна, а управлението на съдържанието става по-лесно и ефективно.

## Conclusion

Case 11-13007 Document 1-1 Filed 07/26/11 Page 11 of 11

<https://localhost:44384>

## **7. Развитие и нововъведения**

Времето, с което разполагахме, беше достатъчно, за да реализираме основните функционалности на приложението за управление на филми. Въпреки че сме постигнали значителен напредък, ние ще продължим да усъвършенстваме приложението, за да го направим още по-функционално и удобно за потребителите. Планираме да добавим следните основни подобрения:

### **7.1. Страница за прожекции в кина:**

- Ще бъде добавена нова страница, която ще предоставя информация за кинотеатри, където се прожектират филмите от нашия каталог. Тази страница ще включва:
- Списък с кинотеатри, актуални за прожекции.
- Информация за адреси, работно време и контакти на кинотеатри.
- Календар или график на прожекциите за отделни филми, за да могат потребителите лесно да намерят най-удобното време за посещение.

### **7.2. Страници за актьори и режисьори:**

- За да предоставим по-пълна информация за екипите зад филмите, планираме да добавим отделни страници, посветени на актьорите и режисьорите. Тези страници ще съдържат:
- Биографии и професионални профили.
- Списък с филмите, в които са участвали.
- Актуални новини и снимки, свързани с тяхната кариера

### **7.3. Добавяне на "Любими филми":**

- Потребителите да могат да добавят филми в списък с "Любими", което ще им позволи бързо да намират и преглеждат филми, които са харесали.

## 8. Заключение

Проектът "Movie Catalog" беше успешно реализиран, като се постигнаха всички поставени цели. Чрез разработката на това уеб приложение успяхме да създадем удобен и функционален инструмент за управление на филми, който позволява на потребителите да търсят, преглеждат и управляват информация за филми, режисьори, жанрове и ревюта. Приложението е изградено с помощта на съвременни технологии като .NET, Entity Framework и ASP.NET MVC, което гарантира неговата стабилност и мащабируемост. Успяхме да интегрираме ключови функционалности като търсене, филтриране, регистрация на потребители и управление на данни, което прави приложението полезно както за обикновени потребители, така и за администратори. Този проект ни позволи да развием уменията си в областта на уеб разработката, работата с бази данни и управлението на проекти. Надяваме се, че приложението ще бъде полезно за всички, които го използват, и че ще продължим да го развиваме в бъдеще.

## 8. Използвана литература

### 8.1 Книги

#### 1. **ASP.NET Core in Action" – Andrew Lock**

- Подробно ръководство за разработка на уеб приложения с ASP.NET Core, което обяснява основни концепции като MVC архитектурата, автентикация, бази данни и работа с API.

#### 2. **"Entity Framework Core in Action" – Jon P. Smith**

- Практически насочена книга за работа с ORM Entity Framework, използван за управление на базата данни в нашето уеб приложение.

## 8.2 Линкове

1.Официална документация на Microsoft за ASP.NET Core и MVC

 <https://learn.microsoft.com/en-us/aspnet/core/>

2.Entity Framework Core – официална документация

 <https://learn.microsoft.com/en-us/ef/core/>

3.Видеоурок: ASP.NET Core MVC Tutorial for Beginners (от IAmTimCorey)

 <https://www.youtube.com/watch?v=fmvcAzHpsk8>

4. Entity Framework Core Code First – пълно ръководство (от dotnet)

 <https://www.youtube.com/watch?v=Ofg7HnKgTsg>

5.ASP.NET Core Web API – Как да изградим REST API (от Les Jackson)

 <https://www.youtube.com/watch?v=pS8OID3yk70>

6.Как да работим с ASP.NET Identity (регистрация и логин система) (от Code Maze)

 <https://www.youtube.com/watch?v=GY1fQH4XWM>

7.Изграждане на уеб приложение с .NET Core и React (от Traversy Media)

 <https://www.youtube.com/watch?v=Fdf5aTYRW0E>

8.Видеоурок за работа с Entity Framework Core Migrations (от Coding Droplets)

 <https://www.youtube.com/watch?v=vS7i4zKsEKI>