РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПРОСМОТРА ФИЛЬМОВ С ПОМОЩЬЮ VISUAL STUDIO

DEVELOPMENT OF APPLICATION FOR WATCHING MOVIES USING VISUAL STUDIO



УДК 004.42

Журавлев Александр Александрович, студент, Уральский федеральный университет им. Б. Н. Ельцина, ИРИТ-РтФ, г. Екатеринбург, ул. Мира, д. 32, e-mail: SanyaProgrammer2503@gmail.com

Zhuravlev A.A. SanyaProgrammer2503@gmail.com

Аннотация

Фильмы играют большую роль в жизни современного человека. Они помогают людям отвлечься от рутинных делать. Для более быстрого поиска интересуемых фильмов можно разработать приложение, позволяющее осуществить пользователю данное действие. Приложение удовлетворяет следующим требованиям: программа должна содержать информацию об актерах, режиссерах, жанрах и фильмах на отдельных страницах; при нажатии на название фильма должен происходить переход на страницу с видео по выбранному фильму. Приложение разработано с помощью программы Visual Studio Community 2019 и платформы ASP.NET Core. Программа может быть полезна для киноманов.

Annotation

Movies play an important role in the life of a modern person. They help people to take their mind off the routine. For a faster search for films of interest, it is possible

to develop an application that allows a user to perform this action. The application meets the following requirements: the program must contain information about actors, directors, genres and movies on separate pages; when a user click on the name of the movie, video for the selected movie is displayed on another page. The application was developed with Visual Studio Community 2019 and ASP.NET Core. The program can be useful for moviegoers

Ключевые слова: разработка, приложение, просмотр фильмов, Visual Studio

Keywords: development, application, movie watching, Visual Studio

Введение

Фильмы играют большую роль в жизни современного человека. Они помогают человеку отвлечься от рутинных делать. Для более быстрого поиска интересуемых фильмов можно разработать приложение, позволяющее осуществить пользователю данное действие.

Цель статьи – с помощью программы Visual Studio и платформы ASP.NET Core разработать приложение для просмотра фильмов, описание которого приведено в разделе «Описание предметной области».

В качестве **материала исследования** выбраны фильмы.

В статье использован **эмпирический метод исследования**, поскольку основной источник результатов – проектирование.

Постановка задачи

Разработать приложение для просмотра фильмов, которое соответствует следующим требованиям:

1. Программа должна содержать информацию об актерах, режиссерах, жанрах и фильмах на отдельных страницах.
2. При нажатии на название фильма должен происходить переход на страницу с видео по выбранному фильму.

Разработка приложения

Для разработки приложения, описанного в разделе «Постановка задачи», выбрана среда Visual Studio Community 2019 [1]. Приложение построено по принципу MVC (Model-View-Controller) [2, 3]. Для хранения информации используется Microsoft SQL Server [4].

База данных состоит из шести таблиц. Код сущностей для каждой таблицы представлен ниже.

1) Actor – актер.

```
namespace DataLayer.Entities.Actors
{
    public class Actor
    {
        public int Id { get; set; } // Идентификационный номер
        public string Name { get; set; } // Имя актера
        public string LastName { get; set; } // Фамилия актера
        public string ImageUrl { get; set; } // Ссылка на фотографию
        public string Country { get; set; } // Страна
        public DateTime BirthDate { get; set; } // Дата рождения
        public string Description { get; set; } // Информация об актере
    }
}
```

2) Director – режиссер.

```
namespace DataLayer.Entities.Directors
{
    public class Director
    {
        public int Id { get; set; } // Идентификационный номер
        public string Name { get; set; } // Имя режиссера
        public string LastName { get; set; } // Фамилия режиссера
        public string ImageUrl { get; set; } // Ссылка на фотографию
        public string Country { get; set; } // Страна
        public DateTime BirthDate { get; set; } // День рождения
```

```
        public string Description { get; set; } // Информация о
        режиссеере
        public List<Movie> Movies { get; set; } // Список фильмов
    }
}
```

3) Genre – жанр фильма.

```
namespace DataLayer.Entities.Genres
{
    public class Genre
    {
        public int Id { get; set; } // Идентификационный номер
        public string Name { get; set; } // Название жанра
        public string Description { get; set; } // Описание жанра
    }
}
```

4) Movie – фильм. Имеет отношение 1 ко многим) с таблицей Director (поскольку режиссер может снимать несколько фильмов, многие ко многим с таблицами Actor (Актер может сниматься в нескольких фильмах, а в одном фильме сниматься несколько актеров) и Genre (Один и тот жанр может относится к множеству фильмов, а в одном фильме иметься несколько жанров). Для обеспечения связей с таблицами Actor и Genre необходимо создание дополнительных таблиц ActorMovieRelation и MovieGenreRelation.

```
namespace DataLayer.Entities.Movies
{
    public class Movie
    {
        public int Id { get; set; } // Идентификационный номер
        public string Name { get; set; } // Название фильма
        public string Country { get; set; } // Страна
        public int Year { get; set; } // Год
        public string Description { get; set; } // Описание фильма
        public string Url { get; set; } // Ссылка на фильм
```

```
        public string ImageUrl { get; set; } // Ссылка на постер
        public Director Director { get; set; } // Внешний ключ к таблице
       Director
        public int DirectorId { get; set; } // Id режиссера
    }
}
```

5) ActorMovieRelation хранит связь между актером и фильмом.

```
namespace DataLayer.Entities.ActorMovieRelations
{
    public class ActorMovieRelation
    {
        public int Id { get; set; } // Идентификационный номер
        public int ActorId { get; set; } // Id актера
        public int MovieId { get; set; } // Id фильма
    }
}
```

6) MovieGenreRelation хранит связь между фильмом и жанром.

```
namespace DataLayer.Entities.MovieGenreRelations
{
    public class MovieGenreRelation
    {
        public int Id { get; set; } // Идентификационный номер
        public int MovieId { get; set; } // Id фильма
        public int GenreId { get; set; } // Id жанра
    }
}
```

Страницы с информацией об актерах, режиссерах, жанрах и фильмах представлены на рисунках 1-4 соответственно.

Онлайн-фильмы Главная страница Фильмы Актеры Режиссеры Жанры

Актеры

Имя или фамилия актера: Найти

№	Актер	Фотография	Страна	Возраст	Фильмы	Информация
1	Леонардо ДиКаприо		США	46	Титаник	американский актёр, продюсер и общественный деятель. Лауреат многочисленных наград, включая премию «Оскар», BAFTA, Премию Гильдии киноактёров США, три премии «Золотой глобус», а также «Серебряного медведя» Берлинского кинофестиваля.
2	Брэд Питт		США	57	Бесславные ублюдки, Бойцовский клуб	американский актёр и кинопродюсер. Лауреат двух премий «Золотой глобус». Обладатель премии «Оскар» как один из продюсеров фильма «12 лет рабства» — победителя в категории «Лучший фильм» на церемонии 2014 года — и за лучшую мужскую роль второго плана в картине «Однажды в Голливуде».

Рисунок 1 – Страница с информацией об актерах

Онлайн-фильмы Главная страница Фильмы Актеры Режиссеры Жанры

Режиссеры

Имя или фамилия режиссера: Найти

№	Режиссер	Фотография	Страна	Возраст	Фильмы	Информация
1	Дэвид Финчер		США	58	Бойцовский клуб	американский кинорежиссёр, продюсер и клипмейкер. Наиболее известен по созданию психологических триллеров, его работы удостоились многочисленных номинаций на премию «Оскар» и «Золотой глобус».
2	Ридли Скотт		Англия	83	Гладиатор	Британский кинорежиссёр и продюсер. Создатель многих прославленных исторических и фантастических фильмов, таких как «Чужой», «Гладиатор», «Царство небесное», «Бегущий по лезвию», «Гангстер», «Совокупность лжи», «Чёрный ястреб», «Марсианин».

Рисунок 2 – Страница с информацией о режиссерах

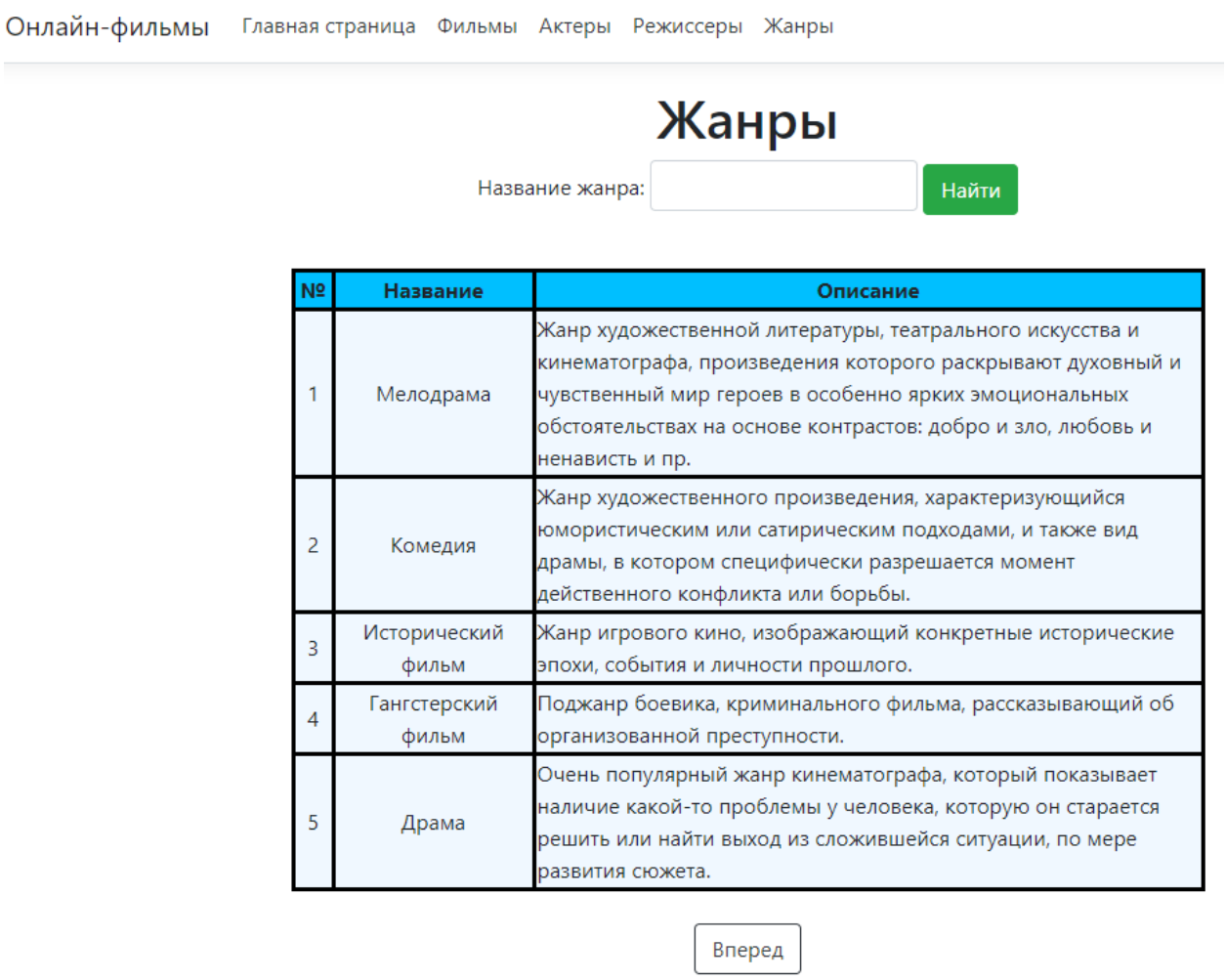


Рисунок 3 – Страница с информацией о жанрах



Рисунок 4 – Страница с информацией о фильмах

На каждой из выше представленных страницах пользователь может осуществлять поиск по названию, а также выполнять переход на другую страницу (пагинацию).

При нажатии на название фильма происходит переход на страницу с видео по выбранному фильму (рисунок 5).



Рисунок 5 – Видео по выбранному фильму

Разработанное приложение может быть полезно для киноманов.

Заключение

В данной статье представлена разработка приложения для просмотра фильмов с помощью среды Visual Studio Community 2019 и платформы ASP.NET Core. В приложении представлена информация об актерах, фильмах, жанрах и фильмах. При нажатии на название фильма происходит переход на страницу с видео по выбранному фильму. Данная программа может быть полезна для людей, увлекающихся кино.

Литература

1. Документация по Visual Studio [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019&preserve-view= true

2. ASP.NET Core MVC [Электронный ресурс]. URL: https://metanit.com/sharp/aspnet5/3.1.php
3. Руководство по Entity Framework Core [Электронный ресурс]. URL: https://metanit.com/sharp/entityframeworkcore/
4. Руководство по MS SQL Server 2019 [Электронный ресурс]. URL: https://metanit.com/sql/sqlserver/
5. Полное руководство по C# 9 и .NET 5 [Электронный ресурс]. URL: https://metanit.com/sharp/tutorial/
6. Дасгупта С. Алгоритмы / С. Дасгупта, Х. Пападимитриу, У. Вазирани; Пер. с англ. под ред. А. Шеня. — М.: МЦНМО, 2014. — 320 С.

Literature

1. Visual Studio Documentation [Electronic resource]. URL: https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019&preserve-view= true
2. ASP.NET Core MVC [Electronic resource]. URL: https://metanit. com/sharp/aspnet5/3.1.php
3. Entity Framework Core Guid [Electronic resource]. URL: https://metanit.com/sharp/entityframeworkcore/
4. MS SQL Server 2019 Guide [Electronic resource]. URL: https://metanit.com/sql/sqlserver/
5. C# 9 и .NET 5 Complete Guid [Electronic resource]. URL: https://metanit.com/sharp/tutorial/
6. Dasgupta S. Algorithms / S. Dasgupta, H. Papadimitriou, U. Wazirani; Translate from English under the editorship of A. Shen. - M: 2014. - 320 P.