

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з дисципліни “Об’єктно -орієнтоване програмування”
<НАЗВА>

Керівник , Ст. викл.

Проф Бондарєв В. М.

Студент гр.

ПЗПІ-22-1 Горбачов Максим

Комісія:

Проф. _____ Бондарєв Н. С.

Ст. викл. _____ Черепанова Ю. Ю.

Ст. викл. _____ Ляпота В. М.

Харків 2023

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*Рівень вищої освіти *перший (бакалаврський)*Дисципліна *Об'єктно-орієнтоване програмування*Спеціальність *121 Інженерія програмного забезпечення*Освітня програма: *Програмна інженерія*Курс 1. Група ПЗП-22 -1. Семестр 2.**ЗАВДАННЯ****на курсовий проект студента****Горбачов Максим Дмитрович**

(Прізвище, Ім'я, По батькові)

1 Тема проекту:

Відео бібліотека

2 Термін здачі студентом закінченого проекту: **“16” - червня - 2023 р.**

3 Вихідні дані до проекту:

Специфікація програми, методичні вказівки до виконання курсової роботи

4 Зміст розрахунково-пояснювальної записки:

Вступ, опис вимог, проектування програми, інструкція користувача, висновки

КАЛЕНДАРНИЙ ПЛАН

| <i>№</i> | <i>Назва етапу</i> | <i>Термін виконання</i> |
|----------|--|----------------------------|
| 1 | Видача теми, узгодження і затвердження теми | 13.02.2023 - 14.03.2023 р. |
| 2 | Формулювання вимог до програми | 15-03-2023 – 30-03-2023 р. |
| 3 | Розробка зберігання та пошуку даних | 01-04-2023 – 11-04-2023 р. |
| 4 | Розробка функцій авторизації, реєстрації | 11-04-2023 – 01-05-2023 р. |
| 5 | Розробка функцій зберігання та завантаження даних | 02-05-2023 – 08-05-2023 р. |
| 6 | Тестування і доопрацювання розробленої програмної системи | 09-05-2023 – 18-05-2023 р. |
| 7 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 19-05-2023 – 25-05-2023 р. |
| 8 | Захист | 05.06.2023 – 16.06.2023 р. |

Студент _____ Горбачов М. Д. _____

(Прізвище, Ім'я, По батькові)

Керівник _____ Бондарєв В. М. _____

(Прізвище, Ім'я, По батькові)

«13»_лютого_____ 2023 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 47 с., 21 рис., 1 додаток, 3 джерел.

ВІДЕО БІБЛІОТЕКА, МОВА ПРОГРАМУВАННЯ C#, КЛАС, ООП.

Метою роботи є розробка програми «Відео бібліотека», програма дозволить робити оренду відео з бібліотеки. Шукати потрібне відео.

У результаті роботи буде отримано додаток, який здатний значно покращити та оптимізувати роботу з паперами в відео бібліотеці.

Зміст

| | |
|---------------------------------|----|
| Вступ | 6 |
| 1 Специфікація програми | 7 |
| 1.1 Функції програми | 7 |
| 1.2 Інтерфейс користувача | 8 |
| 2 Проектна специфікація | 14 |
| 2.1 Об'єктна модель програми | 14 |
| 2.2 Реалізація функцій програми | 15 |
| 2.3 Формат даних | 17 |
| 3 Інструкція користувача | 19 |
| Висновки | 23 |
| Перелік джерел посилання | 24 |
| Додаток А Код програми | 25 |

Вступ

Я створював віконний застосунок, що дозволяє користувачу перелік фільмів чи відео, їх опис, а після цього орендувати дане відео чи фільм. Моя програма може бути використана бібліотекою, щоб користувач орендував щось, прийшов до бібліотеки і отримав те що йому потрібно. У роботі використано принципи об'єктно-орієнтованого програмування.

Програма що я створив, підтримує та взаємодіє з умовною базою даних, бо замість бази даних я використовував текстові файли, виконується додавання, редагування, видалення деяких даних, а також існує пошук між текстом що зберігається, та фільтрування даних за певними характеристиками. В ролі бази даних, використовується текстовий файл ТХТ. Я розробив деяке форматування для даного файлу щоб з ним було зручніше працювати. Інформацію я розділяю а далі вона обробляється і використовується.

Програма автоматично створює для кожного нового користувача його ідентифікаційний номер, питає в користувача його адресу, і має можливість додати орендовану книжку, орендована книжка зберігається після виходу з програми як і ідентифікаційний номер з адресою. Змінити адресу і ідентифікаційний номер в програмі неможливо, а от змінити орендоване відео чи фільм просто.

Підходи об'єктно-орієнтованого програмування дозволяють організувати код краще, також, спрощує реалізацію функціоналу, що дуже важливо, дозволяє побудувати прозору програмну архітектуру, що полегшує процес розробки нових функцій у програмі в майбутньому з мінімальним втручанням у вже написаний програмний код.

Метою роботи є написання програми з можливістю переглядати існуючу в бібліотеці інформацію про фільми чи відео, орендовувати та віддавати фільми, для користувача бібліотеки, за допомогою його ідентифікаційного номера та адреси. Результатом роботи має стати закріплення вивченого з дисципліни "Об'єктно-орієнтоване програмування", а також практика з створення застосунків.

СПЕЦИФІКАЦІЯ ПРОГРАМИ

1.1 Функції програми

Додаток надаватиме користувачу наступні можливості:

- ❖ Автоматичне створення ID користувача
- ❖ Введення адреси яка вимагається для отримання відео
- ❖ Автоматичний вхід до вже створеного акаунту
- ❖ Перегляд інформації про фільми що містяться у бібліотеці
- ❖ Пошук фільму чи відео за назвою чи будь яким іншим параметром.
- ❖ Відображення лише тих, що підходять за результатами пошуку
- ❖ Розширений пошук, що включає в себе типи фільмів
- ❖ Очистка полів пошуку
- ❖ Оренда фільму чи відео, не більше 1 для акаунту
- ❖ Збереження орендованого фільму
- ❖ Видалення фільму з оренди
- ❖ Зміна розмірів інтерфейсу на будь які що захоче користувач
- ❖ Скролінг лише частини інтерфейсу на якому знаходиться інформація про фільми

Усі дані користувачів: ID, адреса, орендований фільм, автоматично зберігаються до локального текстового файлу. Фільтрація, зміна розмірів, додавання чи видалення фільмів, миттєво змінюють інтерфейс.

1.2 Інтерфейс користувача

При запуску програми в 1 раз, користувач побачить, запрошення ввести його адресу, це обов'язково треба зробити, бо без адреси не можливо буде орендувати фільми. Користувач може спробувати не ввести адресу, але в нього буде не доступна оренда в подальшому. Додати адресу можна лише при запуску програми.

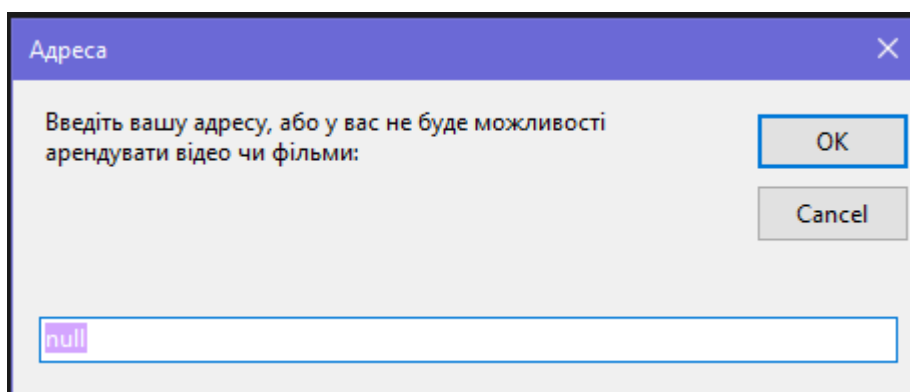


Рисунок 1.1 – Текстове поле введення адреси

Коли користувач ввів адресу він має натиснути «OK», якщо ж користувач хоче лише подивитись які фільми містяться у бібліотеці він може відхилити введення і закрити це вікно, або натиснути «Cancel». Після відхилення користувач буде проінформований.

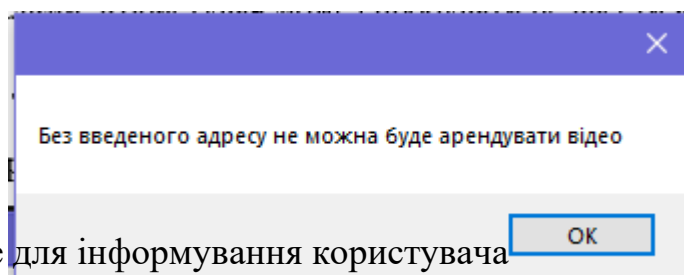


Рисунок 1.2 – Поле для інформування користувача

Якщо ж користувач ввів адресу він перейде до основного вікна, де знаходиться весь функціонал програми. В ситуації коли користувач вже заходив до застосунку і вводив свою адресу він також одразу побачить основне вікно.

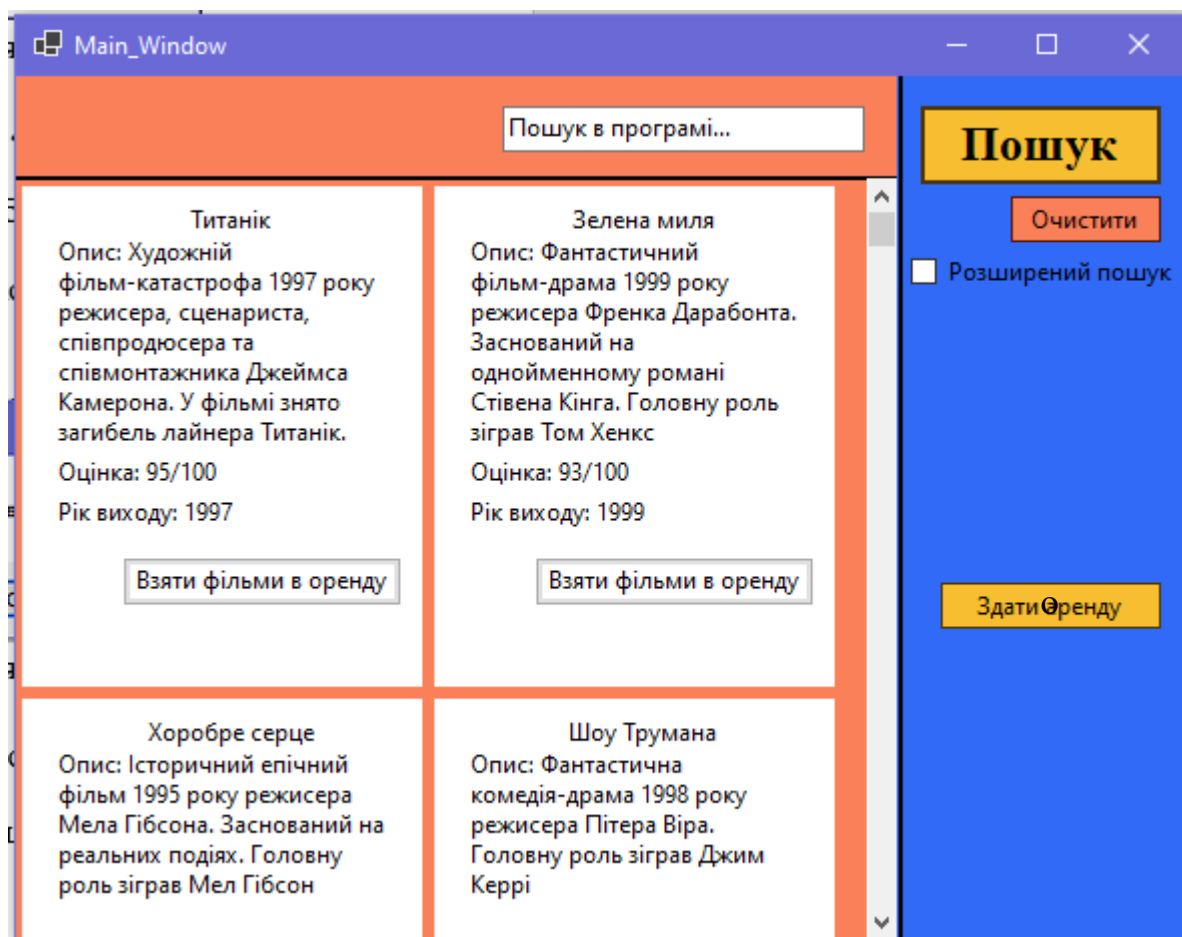


Рисунок 1.3 – Основне вікно програми

В правому нижньому кутку програми можна побачити кнопку здати оренду. Якщо користувач вже орендував якісь фільми, під цією кнопкою буде розташоване текстове поле з назвою орендованого фільму.

Для оренди фільму не треба нічого складного, користувач може зробити це миттєво. Для цього потрібно вибрати фільм із запропонованого списку фільмів по центру програми. Можна скористуватися прокруткою, що допоможе користувачу побачити інші фільми. Якщо користувач побачив

фільм, що йому сподобався, він натискає кнопку що знаходиться на білому фоні під описом.

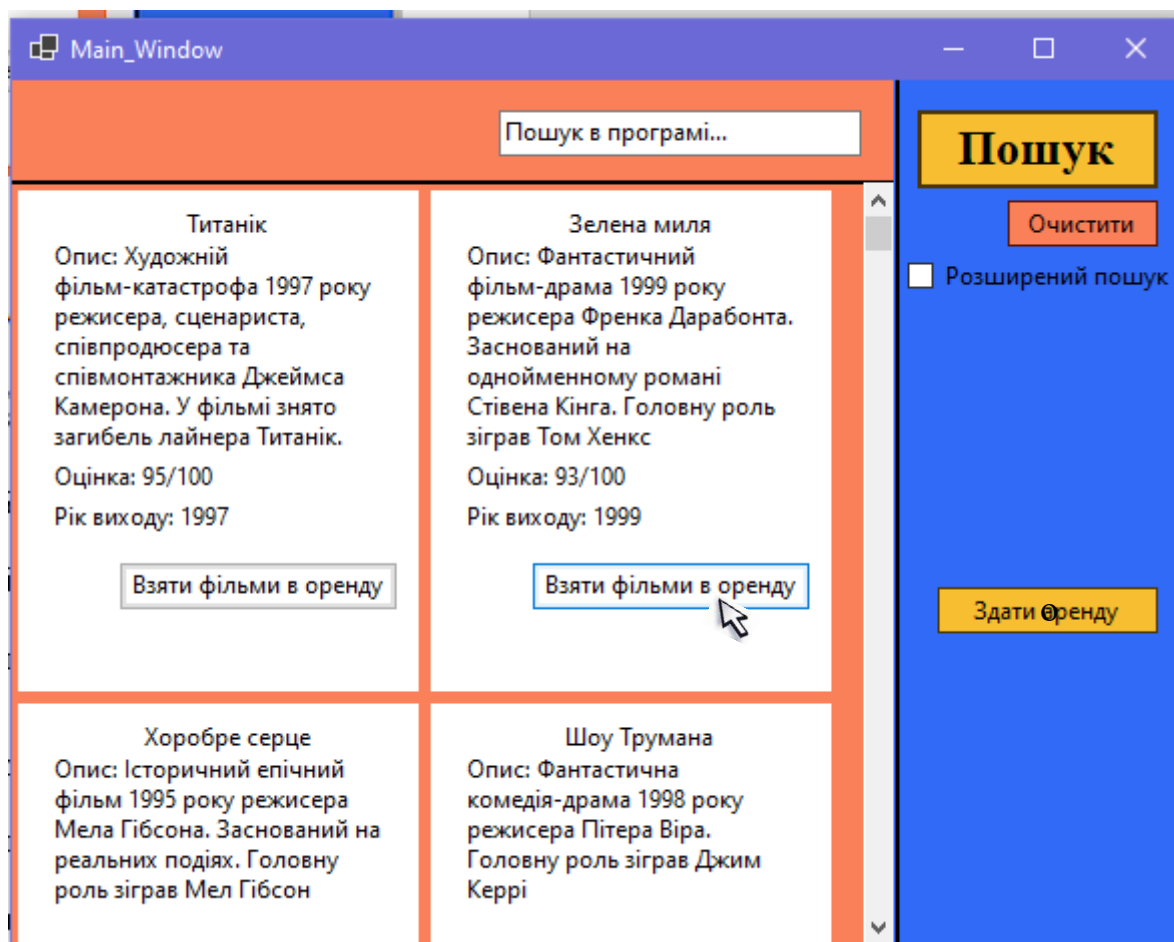


Рисунок 1.4 – Наведення курсору на кнопку оренди фільму

Після натискання на кнопку, фільм моментально з'явиться під кнопкою «Здати оренду». Також фільм що орендовано не відображається у центрі екрану. Це дає змогу вберегти бібліотеку від оренди більшої кількості фільмів ніж міститься в ній. Також користувач одразу знає, що всі фільми що він бачить можливі до оренди.

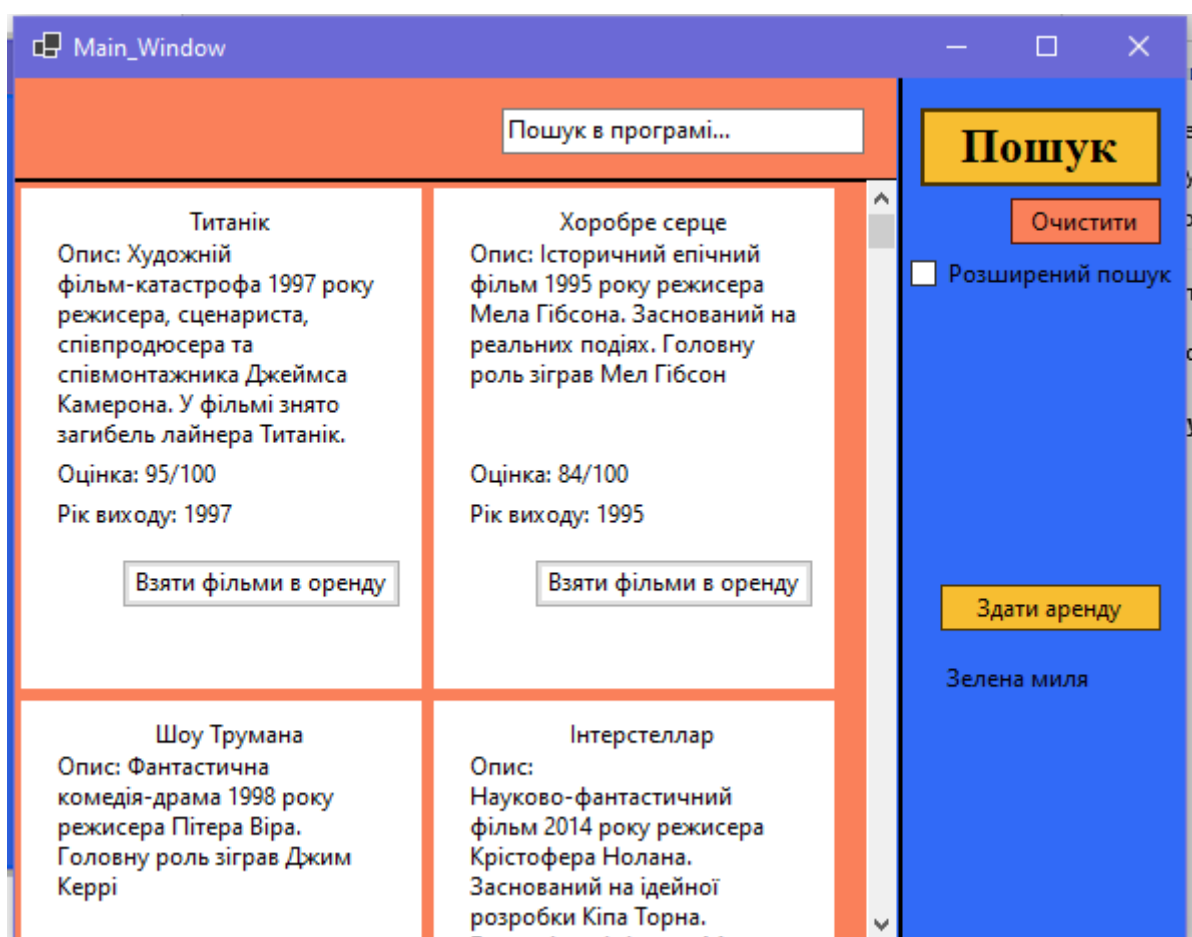


Рисунок 1.5 – Основне вікно програми з орендованим фільмом

Для зручності, якщо користувач знає що він хоче орендувати, можна скористуватися пошуком. В пошук ви можете вписати назву, рік випуску, оцінку, режисера. Після введення це в потрібне поле, і натискання кнопки «Пошук» перед користувачем з'являться фільми що відповідають тексту що він ввів. Текст шукається в програмі за допомогою повного співпадіння, після зменшення та видалення пробілів. Тому якщо ввести слово з помилкою, нажаль програма нічого не знайде, треба бути уважним з цим.

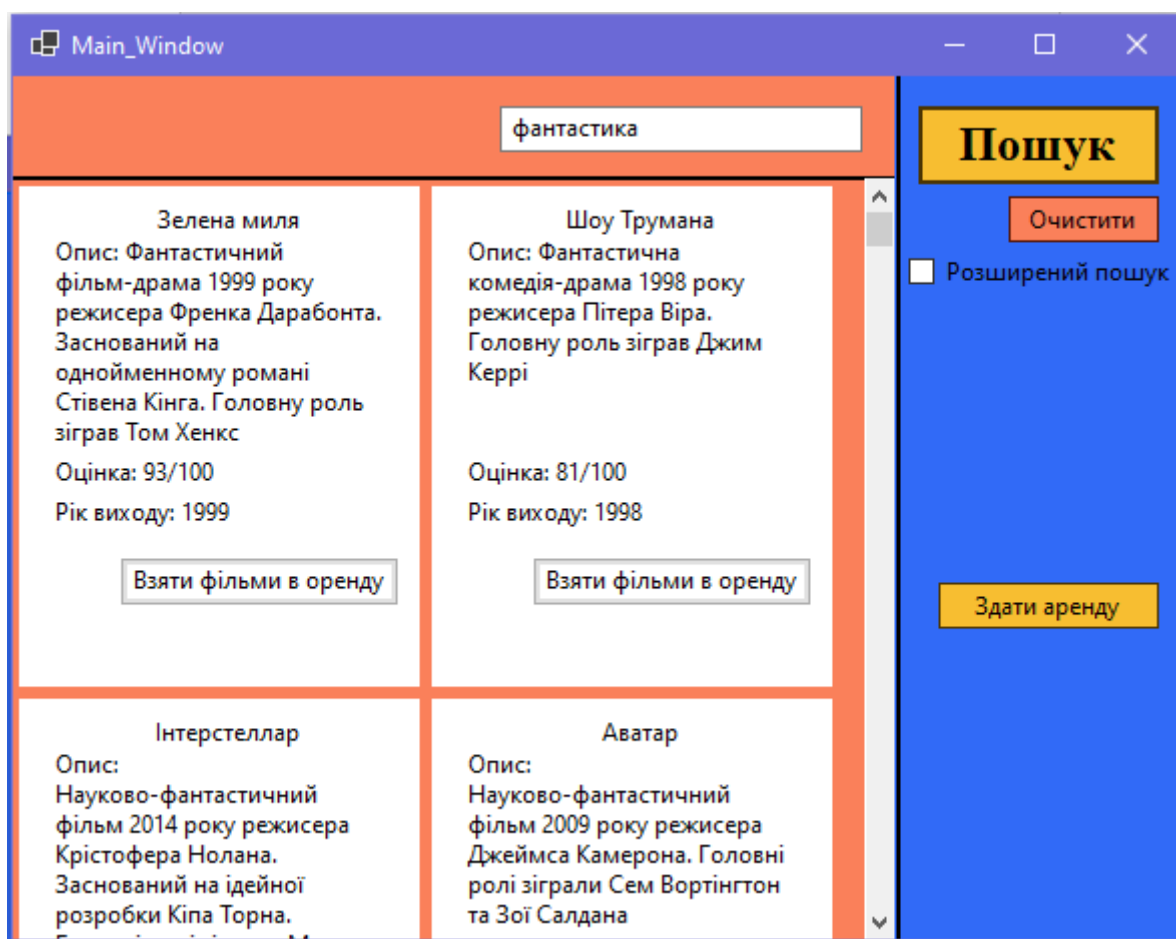


Рисунок 1.6 – Приклад пошуку фільмів.

Існує також розширений пошук, який можна увімкнути після натискання на спеціальну клавішу. Це змінить інтерфейс, і ви побачите додаткові поля, в яких можна вибрати категорію фільму. Це може допомогти тим хто не впевнений у конкретному фільмі, але впевнений у жанрі. Після вибору категорії, автоматично показуються всі фільми що відповідають пошуковому запиту. Нажаль, поєднувати пошукову строку і розширений пошук не вийде, буде шукати або лише пошукова строка, або лише варіанти вибору з розширеного пошуку.

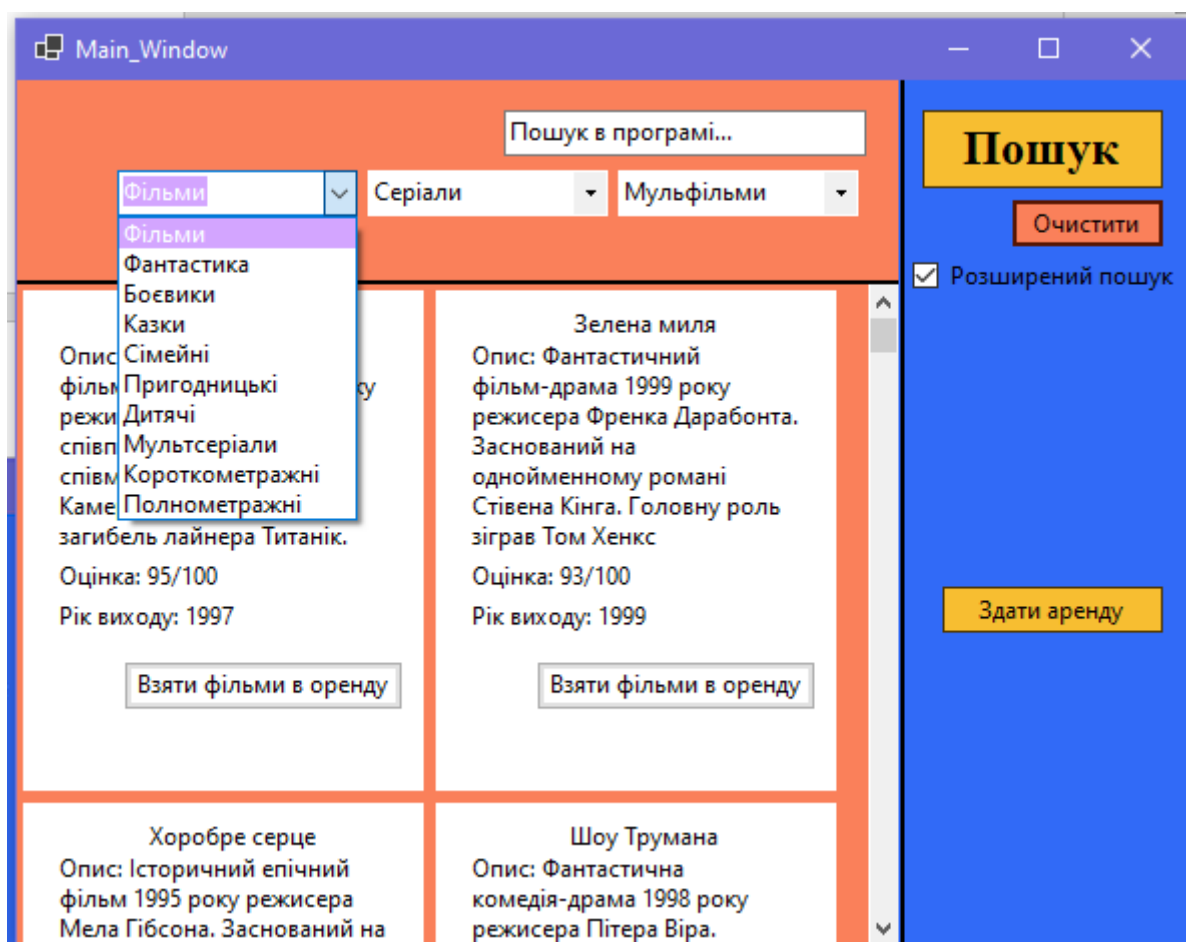


Рисунок 1.7 – Розширений пошук фільмів.

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ

2.1 Об'єктна модель програми

Програма що я розробив, це набір класів, які поділено на моделі, Віндовс-форми, та текстові файли.

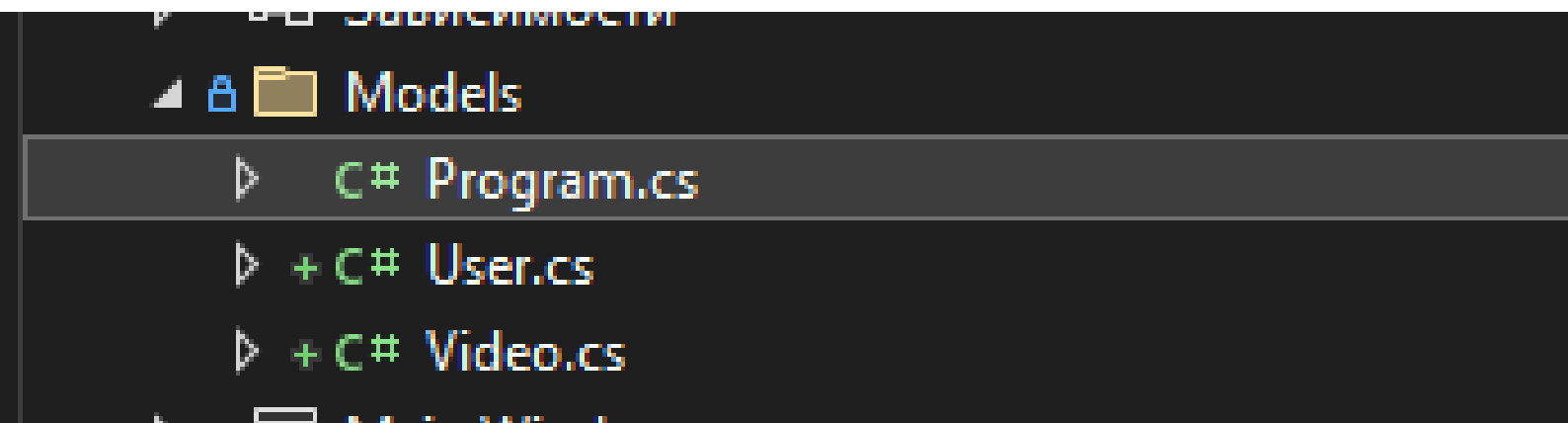


Рисунок 2.1 – Моделі проекту.

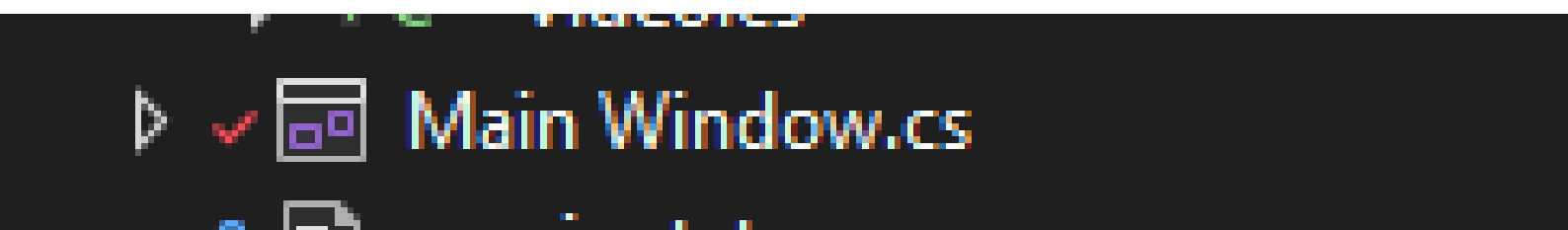


Рисунок 2.2 – Віндовс-форми проекту.



Рисунок 2.3 – Текстові файли проекту.

2.2 Реалізація функцій програми

Перше з чого починається застосунок, це головний клас, в ньому відбувається усі найважливіші функції, ініціалізуються усі компоненти, перевіряються усі дані - це клас Main Window. Запускається він з класу Program.

```
private void InitializeComponent()
{
    movie_combo_box = new ComboBox();
    serial_combo_box = new ComboBox();
    cartoons_combo_box = new ComboBox();
    right_bg = new Panel();
    rented_video = new Label();
    button1 = new Button();
    advanced_search = new CheckBox();
    clear_btn = new Button();
    search_btn = new Button();
    vertical_border = new Panel();
    main_bg = new Panel();
    Video_Panel = new Panel();
    ScrollBarr = new VScrollBar();
    horizon_border = new Panel();
    search_box = new TextBox();
    right_bg.SuspendLayout();
    main_bg.SuspendLayout();
    SuspendLayout();
}
```

Рисунок 2.4 – Створення всіх об'єктів в формі

Після того як усі об'єкти форми їм видаються їх положення, текст, розміри, та інше.

```
//
search_btn.BackColor = Color.FromArgb(247, 190, 49);
search_btn.Cursor = Cursors.Hand;
search_btn.FlatStyle = FlatStyle.Popup;
search_btn.Font = new Font("Times New Roman", 18F, FontStyle.Bold, GraphicsUnit.Point);
search_btn.Location = new Point(12, 15);
search_btn.MaximumSize = new Size(120, 0);
search_btn.MinimumSize = new Size(80, 0);
search_btn.Name = "search_btn";
search_btn.Size = new Size(120, 39);
search_btn.TabIndex = 4;
search_btn.Text = "Пошук";
search_btn.UseVisualStyleBackColor = false;
search_btn.Click += Search_btn_Click;
//
```

Рисунок 2.5 – Надання об'єктам розмірів, тексту, імен, вставлення їх на місце на формі

Далі коли компоненти пройшли ініціалізацію, були виставлені на форму виконуються інші функції, проходить ініціалізація інших класів, таких як User і Video.

```

namespace Video_library.Models
{
    Слайд 9
    internal class User
    {
        Слайд 9
        public int Id { get; set; }
        Слайд 13
        public string Address { get; set; }
        Слайд 12
        public int User_video { get; set; }
        Слайд 6
        public bool Videos_enabel { get; set; }

        Слайд 4
        public User()
        {
            string filePath = "A:\\Проекти C#\\Cursova\\Video_library\\Video_library\\User.txt";

            string text = File.ReadAllText(filePath);

            string[] result = text.Split(';');
            result = result.Where(p => !string.IsNullOrEmpty(p)).ToArray();

            if (result.Length == 0)
            {
                Random random = new Random();
                int randomNumber = random.Next(1, 1000000);

                Address = Microsoft.VisualBasic.Interaction.InputBox("Введіть вашу адресу, або у вас не буде можливості арендувати відео чи фільми:", "Адреса", "null");
                Id = randomNumber;
                if (Address == "null" || Address == "")...
                if (Address != "null")...

                User_video = -1;
            }
            else
            {
                Id = int.Parse(result[0]);
                if (result[1] == "null")
                {
                    if (Address == "null")
                    {
                        MessageBox.Show("Без введення адресу не можна буде арендувати відео");
                        Videos_enabel = false;
                        File.WriteAllText(filePath, $"{Id};{Address};");
                    }
                    else
                    {
                        Videos_enabel = true;
                        Address = result[1];
                        File.WriteAllText(filePath, $"{Id};{Address};");
                    }
                }
            }
            else
            {

```

Рисунок 2.6 – Приклад ініціалізації інших класів, клас User

Реалізація створення об'єктів Video, з текстового файлу, використовуючи метод спліт розділяємо спочатку на масив, після цього розділяємо на допоміжний масив, створюємо двовимірний масив, в якому вже зберігаються об'єкти Video які після цього повертає даний метод:


```

public string[,] My_Init()
{
    string filePath = "A:\\Проекти C#\\Cursova\\Video_library\\Video_library\\movies.txt";
    string text = File.ReadAllText(filePath);

    string[] result = text.Split(' ', '\n');

    result = result.Where(p => !string.IsNullOrEmpty(p)).ToArray();

    List<string[]> nestedArray = new List<string[]>();

    foreach (string part in result)
    {
        string[] components = part.Split(';');
        components = components.Select(c => c.Trim()).Where(c => !string.IsNullOrEmpty(c)).ToArray();

        nestedArray.Add(components);
    }

    string[,] array = new string[nestedArray.Count, nestedArray[0].Length];
    for (int i = 0; i < nestedArray.Count; i++)
    {
        for (int j = 0; j < nestedArray[i].Length; j++)
        {
            array[i, j] = nestedArray[i][j];
        }
    }

    return array;
}

```

Рисунок 2.7 – Реалізація створення об'єктів Video, з текстового файлу

2.3 Формат даних

Дані в мене зберігаються в звичайних тестових файлах, данні в них

зберігаються за допомогою звичайних строк. Я використовую 2 текстових файли, 1 для збереження описів відео, інший для конкретного юзера.

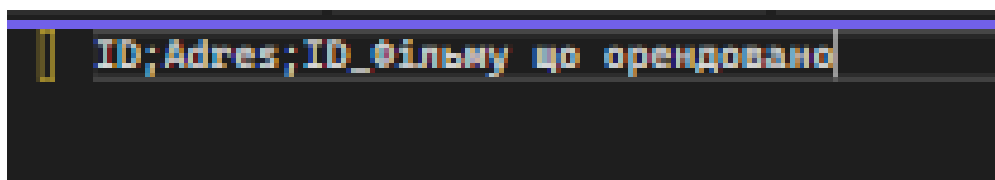


Рисунок 2.8 – Вигляд в якому збережено дані юзера

```
{Титанік;Художній фільм-катастрофа 1997 року режисера, сценариста, співпродюсера та співмонтажника Джеймса Камерона. У фільмі знято за  
{Зелена миля;Фантастичний фільм-драма 1999 року режисера Френка Дарабонта. Заснований на однойменному романі Стівена Кінга. Головну ро  
{Хоробре серце;Історичний епічний фільм 1995 року режисера Мела Гібсона. Заснований на реальних подіях. Головну роль зіграв Мел Гібсон;  
{Вою Трумана;Фантастична комедія-драма 1998 року режисера Пітера Віра. Головну роль зіграв Джим Керрі;1998;Драма, Комедія, Фантастика;  
{Інтерстеллар;Науково-фантастичний фільм 2014 року режисера Крістофера Нолана. Заснований на ідейної розробки Кіпа Торна. Головні ролі  
{Форест Гамп;Драматична картина 1994 року режисера Роберта Земекіса. Заснована на романі Вінстона Грума. Головну роль зіграв Том Хенкс;  
{Аватар;Науково-фантастичний фільм 2009 року режисера Джеймса Камерона. Головні ролі зіграли Сем Вортінгтон та Зої Салдана;2009;Драма,
```

Рисунок 2.8 – Вигляд в якому збережено дані фільмів

Такі дані не тяжко перетворювати на об'єкти, тому мені було зручно їх використати. З мінусів такого збереження це уязвимість даних, і можливість їх підробки.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Робота з програмою

Запустивши програму користувач бачить текстове поле введення адреси.

Користувач має ввести свою адресу щоб в подальшому він міг користуватися орендою фільмів. Вигляд цього текстового поля:

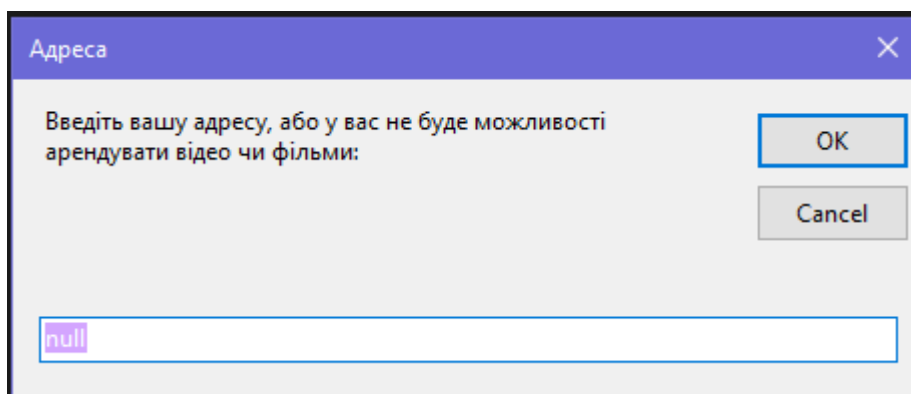


Рисунок 3.1 – Текстове поле введення адреси

Якщо ж все зроблено вірно користувач побачить головне вікно. Є можливість переглянути існуючі в бібліотечці, та доступні для оренди фільми.

Вибрати фільм що сподобався, натиснути кнопку «взяти фільм в оренду»

Після даної маніпуляції фільм буде додано до списку що вже орендував користувач, в правому нижньому кутку.

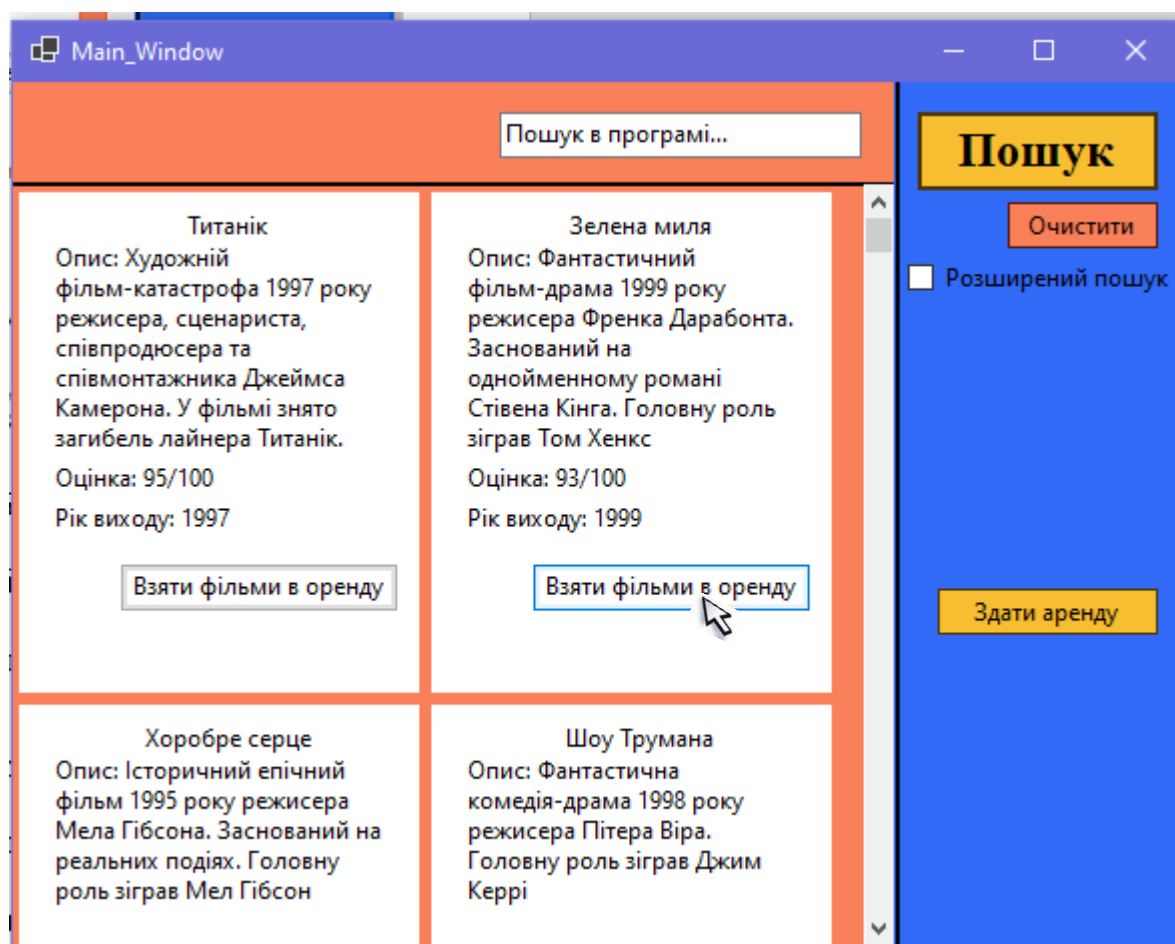


Рисунок 3.2 – Наведення курсору на кнопку оренди фільму

Для здачі орендованого фільму достатньо натиснути кнопку «Здати аренду»

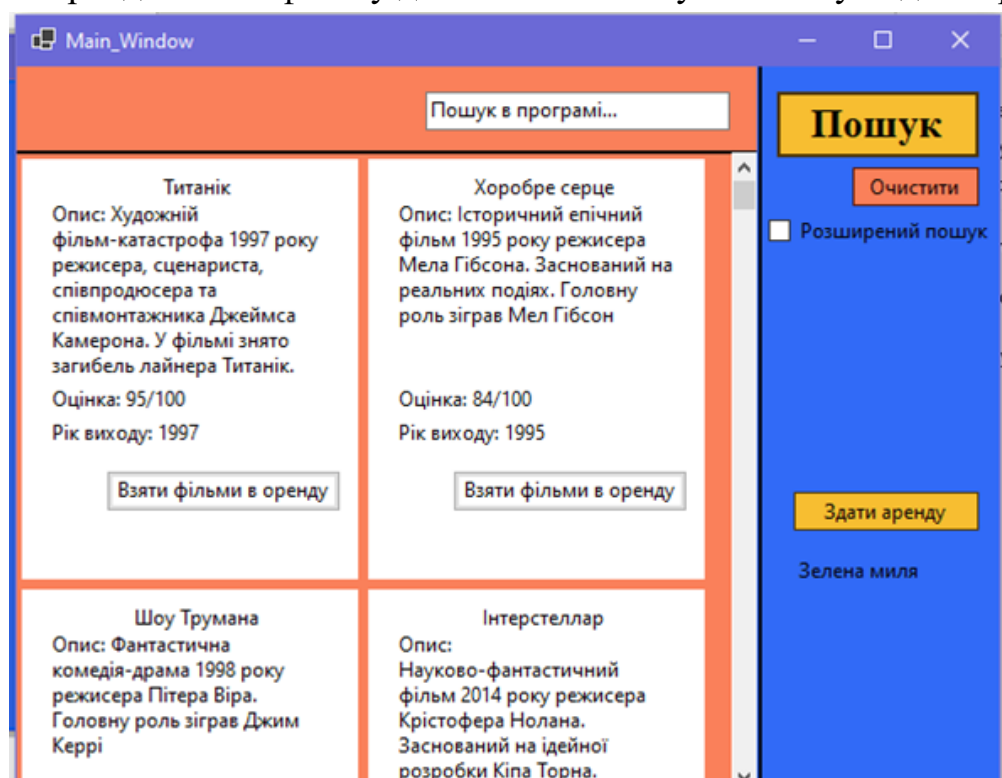


Рисунок 3.3 – Основне вікно програми з орендованим фільмом

Після натискання фільм зникне з оренди і можна буде додати новий.

Для пошуку в програмі можна використовувати текстове поле з текстом

«Пошук в програмі...» Коли текст введено після натиску кнопки «Пошук»

буде показано фільми що відповідають шуканому тексту.

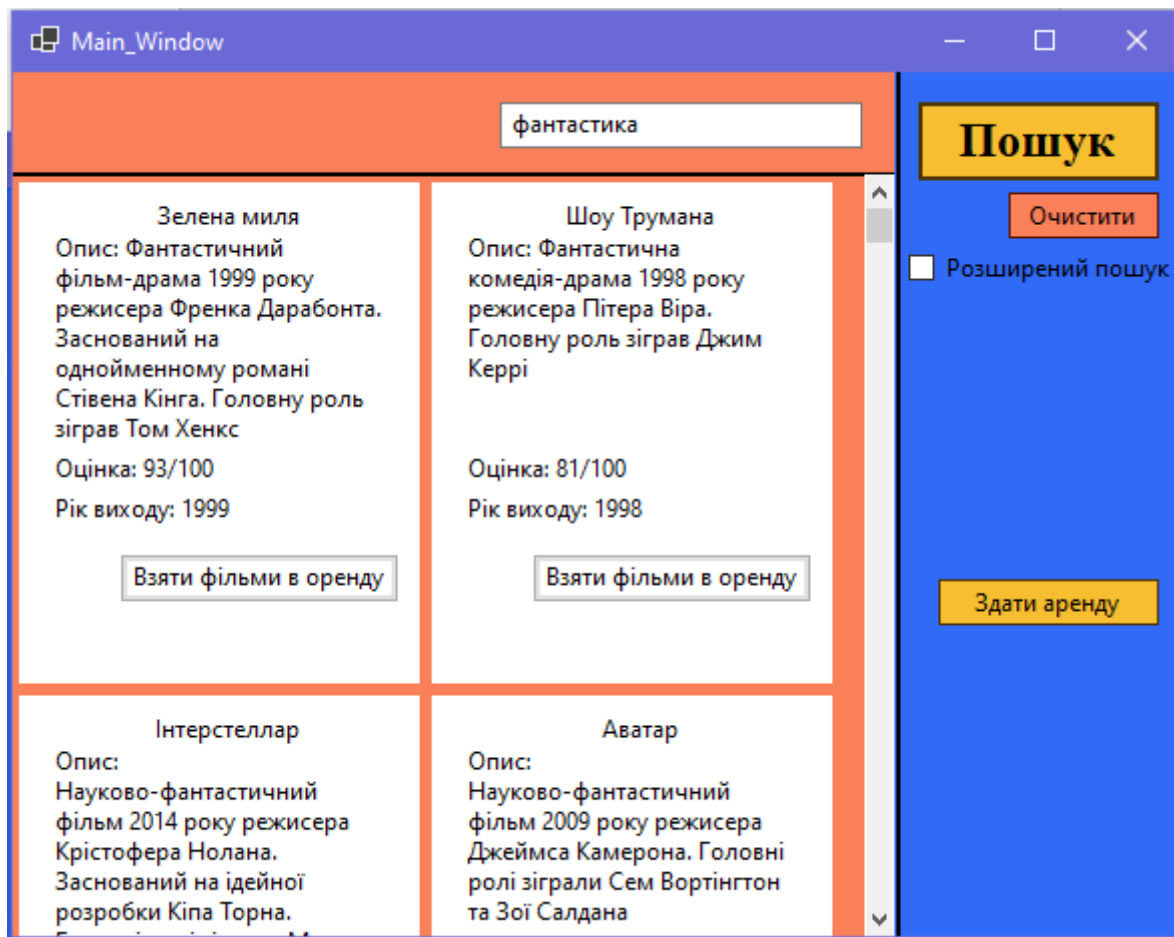


Рисунок 3.4 – Приклад пошуку фільмів.

Натиснувши «Розширений пошук» верхня частина розшириться і з'являться деякі категорії для вибору. Можна вибрати деяку з категорій, після чого автоматично відбудеться пошук фільмів що відповідають даній категорії і їх буде показано.

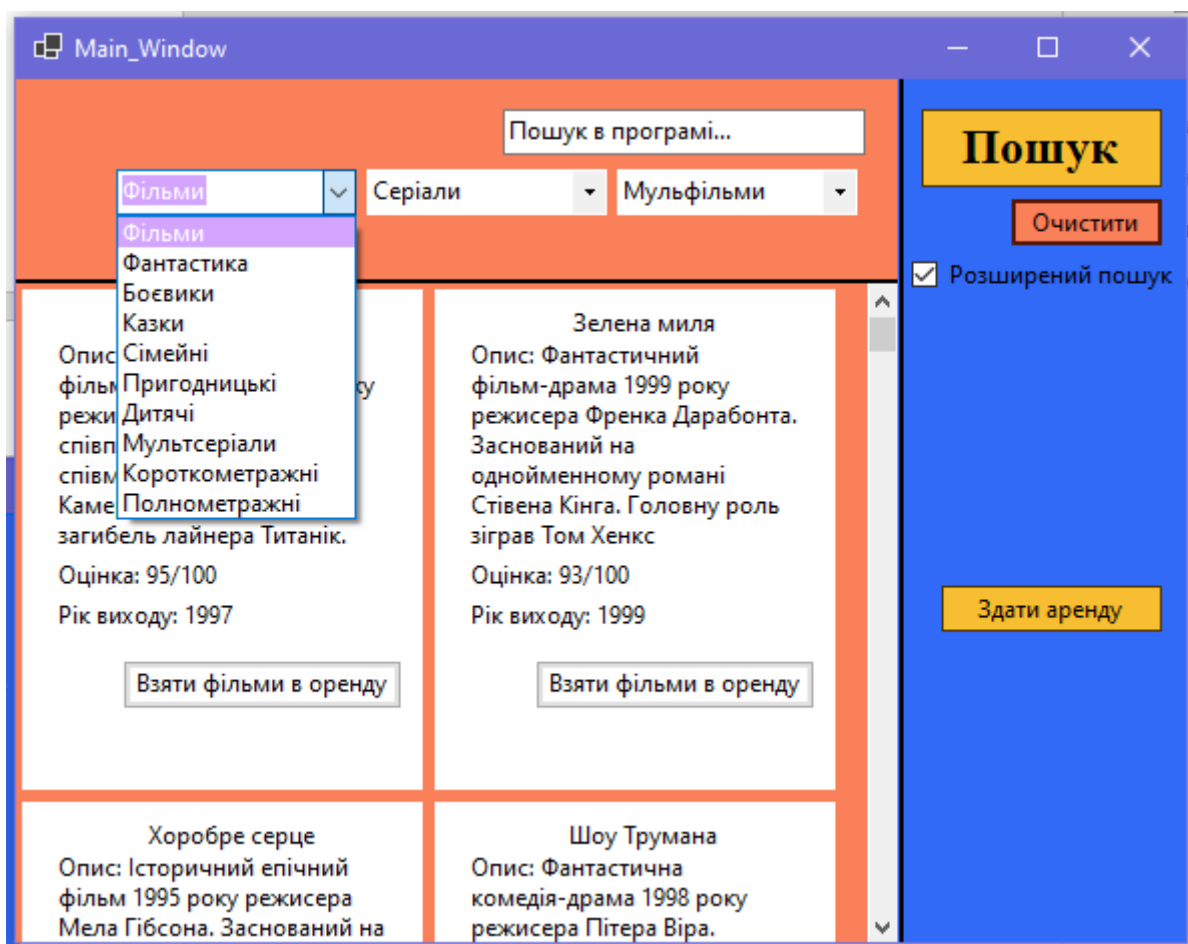


Рисунок 3.5 – Розширений пошук фільмів.

Після пошуку можна скористатися кнопкою «Очистити» це очистить текстові поля і вибір категорій.

ВИСНОВКИ

Під час виконання курсової роботи була розроблена програма «Відео бібліотека», програма дозволяє робити оренду відео з бібліотеки.

У результаті виконання курсової роботи було отримано додаток, який здатний значно покращити та оптимізувати роботу з паперами в відео бібліотеці.

Користуватися додатком можна з комп'ютера. Дані користувача прив'язано окремо для кожного нового девайсу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. С# / BestProg. Програмування: теорія та практика. URL:
https://www.bestprog.net/uk/sitemap_ua/c-3/ (дата звернення: 30.05.2022)
2. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. К.: ДП «УкрНДНЦ», 2016. 26 с.
3. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. К.: ДП «УкрНДНЦ», 2016. – 16 с.

ДОДАТОК А

Код програми

```
namespace Video_library.Models
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            ApplicationConfiguration.Initialize();
            Application.Run(new Main_Window());
        }
    }
}

namespace Video_library
{
    partial class Main_Window
    {
        private System.ComponentModel.IContainer components = null;
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code
        private void InitializeComponent()
        {

```

```

movie_combo_box = new ComboBox();
serial_combo_box = new ComboBox();
cartoons_combo_box = new ComboBox();
right_bg = new Panel();
rented_video = new Label();
button1 = new Button();
advanced_search = new CheckBox();
clear_btn = new Button();
search_btn = new Button();
vertical_border = new Panel();
main_bg = new Panel();
Video_Panel = new Panel();
ScrollBar = new VScrollBar();
horizon_border = new Panel();
search_box = new TextBox();
right_bg.SuspendLayout();
main_bg.SuspendLayout();
SuspendLayout();
//
// movie_combo_box
//
movie_combo_box.Anchor = AnchorStyles.Top | AnchorStyles.Right;
movie_combo_box.CausesValidation = false;
movie_combo_box.FormattingEnabled = true;
movie_combo_box.Items.AddRange(new object[] { "Фільми",
"Фантастика", "Боевики", "Казки", "Сімейні", "Пригодницькі", "Дитячі",
"Мультсеріали", "Короткометражні", "Повнометражні" });
movie_combo_box.Location = new Point(50, 45);
movie_combo_box.Name = "movie_combo_box";
movie_combo_box.Size = new Size(120, 23);
movie_combo_box.TabIndex = 0;
movie_combo_box.Text = "Фільми";
movie_combo_box.Visible = false;
movie_combo_box.SelectedIndexChanged +=
ComboBox_SelectedIndexChanged;
movie_combo_box.MouseLeave += ComboBox_MouseLeave;
movie_combo_box.MouseHover += ComboBox_MouseHover;

```

```

//
// serial_combo_box
//
serial_combo_box.Anchor = AnchorStyles.Top | AnchorStyles.Right;
serial_combo_box.FlatStyle = FlatStyle.Popup;
serial_combo_box.FormattingEnabled = true;
serial_combo_box.Items.AddRange(new object[] { "Серіали",
"Фантастика", "Боевики", "Казки", "Сімейні", "Пригодницькі", "Дитячі",
"Мультсеріали", "Короткометражні", "Полнометражні" });
serial_combo_box.Location = new Point(175, 45);
serial_combo_box.Name = "serial_combo_box";
serial_combo_box.RightToLeft = RightToLeft.No;
serial_combo_box.Size = new Size(120, 23);
serial_combo_box.TabIndex = 1;
serial_combo_box.Text = "Серіали";
serial_combo_box.Visible = false;
serial_combo_box.SelectedIndexChanged +=
ComboBox_SelectedIndexChanged;
serial_combo_box.MouseLeave += ComboBox_MouseLeave;
serial_combo_box.MouseHover += ComboBox_MouseHover;
//
// cartoons_combo_box
//
cartoons_combo_box.Anchor = AnchorStyles.Top |
AnchorStyles.Right;
cartoons_combo_box.FlatStyle = FlatStyle.Popup;
cartoons_combo_box.FormattingEnabled = true;
cartoons_combo_box.Items.AddRange(new object[] { "Мультфільми",
"Фантастика", "Боевики", "Казки", "Сімейні", "Пригодницькі", "Дитячі",
"Мультсеріали", "Короткометражні", "Полнометражні" });
cartoons_combo_box.Location = new Point(300, 45);
cartoons_combo_box.Name = "cartoons_combo_box";
cartoons_combo_box.Size = new Size(120, 23);
cartoons_combo_box.TabIndex = 2;
cartoons_combo_box.Text = "Мультфільми";
cartoons_combo_box.Visible = false;
cartoons_combo_box.SelectedIndexChanged +=
ComboBox_SelectedIndexChanged;

```

```

cartoons_combo_box.MouseLeave += ComboBox_MouseLeave;
cartoons_combo_box.MouseHover += ComboBox_MouseHover;
//
// right_bg
//
right_bg.BackColor = Color.FromArgb(49, 106, 247);
right_bg.Controls.Add(rented_video);
right_bg.Controls.Add(button1);
right_bg.Controls.Add(advanced_search);
right_bg.Controls.Add(clear_btn);
right_bg.Controls.Add(search_btn);
right_bg.Dock = DockStyle.Right;
right_bg.Location = new Point(440, 0);
right_bg.Margin = new Padding(0);
right_bg.Name = "right_bg";
right_bg.Size = new Size(144, 431);
right_bg.TabIndex = 3;
right_bg.Click += Disfocus;
//
// rented_video
//
rented_video.AutoSize = true;
rented_video.Location = new Point(22, 292);
rented_video.Name = "rented_video";
rented_video.Size = new Size(13, 15);
rented_video.TabIndex = 8;
rented_video.Text = "1";
//
// button1
//
button1.BackColor = Color.FromArgb(247, 190, 49);
button1.BackgroundImageLayout = ImageLayout.Center;
button1.FlatStyle = FlatStyle.Popup;
button1.Location = new Point(22, 253);
button1.Name = "button1";
button1.Size = new Size(110, 23);

```

```

button1.TabIndex = 7;
button1.Text = "Здати аренду";
button1.UseVisualStyleBackColor = false;
button1.Click += Submit_arend;
//
// advenced_search
//
advenced_search.AutoSize = true;
advenced_search.CheckAlign = ContentAlignment.TopLeft;
advenced_search.Cursor = Cursors.Hand;
advenced_search.Location = new Point(7, 89);
advenced_search.Name = "advenced_search";
advenced_search.Size = new Size(137, 19);
advenced_search.TabIndex = 6;
advenced_search.Text = "Розширений пошук";
advenced_search.UseVisualStyleBackColor = true;
advenced_search.CheckedChanged += Advenced_search_CheckedChanged;
//
// clear_btn
//
clear_btn.BackColor = Color.FromArgb(250, 128, 90);
clear_btn.BackgroundImageLayout = ImageLayout.Center;
clear_btn.FlatStyle = FlatStyle.Popup;
clear_btn.Location = new Point(57, 60);
clear_btn.Name = "clear_btn";
clear_btn.Size = new Size(75, 23);
clear_btn.TabIndex = 5;
clear_btn.Text = "Очистити";
clear_btn.UseVisualStyleBackColor = false;
clear_btn.Click += Clear_Click;
//
// search_btn
//
search_btn.BackColor = Color.FromArgb(247, 190, 49);
search_btn.Cursor = Cursors.Hand;
search_btn.FlatStyle = FlatStyle.Popup;

```

```

        search_btn.Font = new Font("Times New Roman", 18F,
FontStyle.Bold, GraphicsUnit.Point);

        search_btn.Location = new Point(12, 15);
        search_btn.MaximumSize = new Size(120, 0);
        search_btn.MinimumSize = new Size(80, 0);
        search_btn.Name = "search_btn";
        search_btn.Size = new Size(120, 39);
        search_btn.TabIndex = 4;
        search_btn.Text = "Пошук";
        search_btn.UseVisualStyleBackColor = false;
        search_btn.Click += Search_btn_Click;
        //
        // vertical_border
        //
        vertical_border.Anchor = AnchorStyles.Top | AnchorStyles.Bottom |
AnchorStyles.Right;

        vertical_border.BackColor = Color.Black;
        vertical_border.Location = new Point(441, 0);
        vertical_border.Name = "vertical_border";
        vertical_border.Size = new Size(2, 432);
        vertical_border.TabIndex = 6;
        //
        // main_bg
        //
        main_bg.Anchor = AnchorStyles.Top | AnchorStyles.Bottom |
AnchorStyles.Left | AnchorStyles.Right;

        main_bg.BackColor = Color.FromArgb(250, 128, 90);
        main_bg.Controls.Add(Video_Panel);
        main_bg.Controls.Add(ScrollBarr);
        main_bg.Controls.Add(horizon_border);
        main_bg.Controls.Add(search_box);
        main_bg.Controls.Add(cartoons_combo_box);
        main_bg.Controls.Add(serial_combo_box);
        main_bg.Controls.Add(movie_combo_box);
        main_bg.Location = new Point(0, 0);
        main_bg.Margin = new Padding(0);
        main_bg.MinimumSize = new Size(0, 480);

```

```

main_bg.Name = "main_bg";
main_bg.Size = new Size(440, 480);
main_bg.TabIndex = 6;
main_bg.Click += Disfocus;
//
// Video_Panel
//
Video_Panel.Anchor = AnchorStyles.Top | AnchorStyles.Bottom |
AnchorStyles.Left | AnchorStyles.Right;
Video_Panel.BackColor = Color.FromArgb(250, 128, 90);
Video_Panel.Location = new Point(0, 52);
Video_Panel.Margin = new Padding(0);
Video_Panel.Name = "Video_Panel";
Video_Panel.Size = new Size(425, 379);
Video_Panel.TabIndex = 9;
//
// ScrollBarr
//
ScrollBarr.Anchor = AnchorStyles.Top | AnchorStyles.Bottom |
AnchorStyles.Right;
ScrollBarr.Location = new Point(425, 51);
ScrollBarr.Name = "ScrollBarr";
ScrollBarr.Size = new Size(15, 380);
ScrollBarr.TabIndex = 8;
ScrollBarr.Scroll += VScrollBar_Scroll;
//
// horizon_border
//
horizon_border.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;
horizon_border.BackColor = Color.Black;
horizon_border.Location = new Point(-26, 50);
horizon_border.Name = "horizon_border";
horizon_border.Size = new Size(469, 2);
horizon_border.TabIndex = 7;
horizon_border.Click += Disfocus;
//

```

```

// search_box
//
search_box.Anchor = AnchorStyles.Top | AnchorStyles.Right;
search_box.Location = new Point(243, 15);
search_box.MaximumSize = new Size(250, 0);
search_box.Name = "search_box";
search_box.Size = new Size(181, 23);
search_box.TabIndex = 3;
search_box.Text = "Пошук в програмі...";
search_box.Enter += Search_box_Enter;
search_box.Leave += Search_box_Leave;
//
// Main_Window
//
AutoScaleDimensions = new.SizeF(7F, 15F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(584, 431);
Controls.Add(main_bg);
Controls.Add(vertical_border);
Controls.Add(right_bg);
MinimumSize = new Size(600, 0);
Name = "Main_Window";
Text = "Main_Window";
SizeChanged += Main_Window_SizeChanged;
Click += Disfocus;
right_bg.ResumeLayout(false);
right_bg.PerformLayout();
main_bg.ResumeLayout(false);
main_bg.PerformLayout();
ResumeLayout(false);
}

#endregion

private ComboBox movie_combo_box;
private ComboBox serial_combo_box;

```



```

        private ComboBox cartoons_combo_box;
        private Panel right_bg;
        private Button search_btn;
        private Panel main_bg;
        private TextBox search_box;
        private Button clear_btn;
        private Panel vertical_border;
        private Panel horizon_border;
        private CheckBox advenced_search;
        private VScrollBar ScrollBarr;
        private Panel Video_Panel;
        private Button button1;
        private Label rented_video;
    }
} using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using static System.Runtime.InteropServices.JavaScript.JSType;
using System.Drawing.Drawing2D;
using System.Diagnostics.Metrics;
using System.Data.Common;
using Video_library.Models;

namespace Video_library
{
    public partial class Main_Window : Form
    {
        public string string_to_find { get; private set; }
        public bool Rented { get; set; }
    }
}

```

```

public Main_Window()
{
    InitializeComponent();
    Rented = false;
    User user = new User();
    string_to_find = " ";
    GET_Video();
    if (user.User_video >= 0)
    {
        Rented = true;
    }
}

private void GET_Video()
{
    Video_Panel.Controls.Clear();
    Videos videos = new Videos();
    Video[] videoArray = videos.GetVideos();
    User user = new User();

    string[,] videoMatrix = new string[videoArray.Length, 7];

    for (int i = 0; i < videoArray.Length; i++)
    {
        videoMatrix[i, 0] = videoArray[i].Name;
        videoMatrix[i, 1] = videoArray[i].Description;
        videoMatrix[i, 2] = videoArray[i].Year.ToString();
        videoMatrix[i, 3] = videoArray[i].Genre;
        videoMatrix[i, 4] = videoArray[i].Rating.ToString();
        videoMatrix[i, 5] = videoArray[i].Country;
        videoMatrix[i, 6] = videoArray[i].ID.ToString();
    }

    int Matrixrows = videoMatrix.GetLength(0);
    int Matrixcolumns = videoMatrix.GetLength(1);

    if (user.User_video >= 0)
    {

```

```

        Rented = true;
    }
    else
    {
        Rented = false;
    }

    int element_count = videoArray.Length;
    int columns = (int)Math.Floor((double)Video_Panel.Width / 205);
    int rows = (int)Math.Ceiling((double)element_count / columns);
    if (rows > 0)
    {
        TableLayoutPanel tableLayout = new TableLayoutPanel();

        tableLayout.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;
        tableLayout.Width = Video_Panel.Width;
        tableLayout.Height = 260 * rows;
        ScrollBarr.Value = 0;
        ScrollBarr.Maximum = 255 * rows;

        tableLayout.RowCount = rows;
        tableLayout.ColumnCount = columns;

        int added_count = 0;

        for (int i = 0; i < element_count; i++)
        {
            if (user.User_video != i)
            {
                System.Windows.Forms.Panel panel = new
System.Windows.Forms.Panel();

                for (int j = 0; j < Matrixcolumns; j++)
                {
                    if (string_to_find == null || string_to_find == "
" || videoMatrix[i, j].ToLower().Replace(" ",
"").Contains(string_to_find.ToLower().Replace(" ", "")))

```

```

{
    int row = added_count / columns;
    int column = added_count % columns;
    panel.Name = i.ToString();
    panel.Size = new Size(200, 250);

    System.Windows.Forms.Label textLabel = new
System.Windows.Forms.Label();
    textLabel.Text =
videoArray[i].Name.ToString();
    textLabel.TextAlign =
ContentAlignment.MiddleCenter;
    textLabel.Location = new
System.Drawing.Point(55, 5);
    panel.Controls.Add(textLabel);

    System.Windows.Forms.Label textLabel2 = new
System.Windows.Forms.Label();
    textLabel2.Text = $"Опис:
{videoArray[i].Description.ToString()}";
    textLabel2.TextAlign =
ContentAlignment.TopLeft;
    textLabel2.Location = new
System.Drawing.Point(15, 25);
    textLabel2.Size = new Size(170, 110);
    panel.Controls.Add(textLabel2);

    System.Windows.Forms.Label textLabel3 = new
System.Windows.Forms.Label();
    textLabel3.Text = $"Оцінка:
{videoArray[i].Rating.ToString()}/100";
    textLabel3.TextAlign =
ContentAlignment.TopLeft;
    textLabel3.Location = new
System.Drawing.Point(15, 135);
    textLabel3.Size = new Size(170, 20);
    panel.Controls.Add(textLabel3);

    System.Windows.Forms.Label textLabel4 = new
System.Windows.Forms.Label();

```

```

        textLabel4.Text = $"Рік виходу:
{videoArray[i].Year.ToString()}";

        textLabel4.TextAlign =
ContentAlignment.TopLeft;

        textLabel4.Location = new
System.Drawing.Point(15, 155);

        textLabel4.Size = new Size(170, 20);
        panel.Controls.Add(textLabel4);

        System.Windows.Forms.Button Button = new
System.Windows.Forms.Button();

        Button.Text = $"Взяти фільми в оренду";
        Button.TextAlign = ContentAlignment.TopLeft;
        Button.Location = new
System.Drawing.Point(50, 185);

        Button.Size = new Size(140, 25);
        Button.Name = $"{i};{videoArray[i].Name}";
        Button.Click += Add_Movie;
        Button.Enabled = false;
        if (user.Videos_enabel)
        {
            Button.Enabled = true;
        }
        panel.Controls.Add(Button);

        panel.BackColor = Color.White;
        added_count++;

        tableLayout.Controls.Add(panel, column, row);
        break;
    }
}

}

Video_Panel.Controls.Add(tableLayout);

if (Rented)
{

```

```

        rented_video.Text =
${videoArray[user.User_video].Name}";
    }
    else
    {
        rented_video.Text = string.Empty;
    }
}

private void Search_btn_Click(object sender, EventArgs e)
{
    string name_of = search_box.Text;
    if (name_of != "Пошук в програмі...")
    {
        string_to_find = name_of.ToLower().Replace(" ", "");
        GET_Video();
    }
    else
    {
        string_to_find = " ";
        GET_Video();
    }
}

private void Add_Movie(object sender, EventArgs e)
{
    User user = new User();

    System.Windows.Forms.Button clickedButton = sender as
System.Windows.Forms.Button;

    string buttonName = clickedButton.Name;
    string[] arr = buttonName.Split(";");

    if (user.User_video >= 0)
    {
        MessageBox.Show($"Ви не можете взяти в оренду більше 1
відео!");
    }
    else

```

```

        {
            rented_video.Text = arr[1];
            user.Add_Video(int.Parse(arr[0]), arr[1]);
            GET_Video();
        }
    }

    private void ComboBox_SelectedIndexChanged(object sender, EventArgs
e)
    {
        System.Windows.Forms.ComboBox type_video =
(System.Windows.Forms.ComboBox) sender;

        string_to_find = " ";

        if (type_video.Text != "Фільми" || type_video.Text != "Серіали"
|| type_video.Text != "Мультфільми")
        {
            string_to_find = type_video.Text;
        }

        GET_Video();
    }

    private void ComboBox_MouseHover(object sender, EventArgs e)
    {
        System.Windows.Forms.ComboBox comboBox =
(System.Windows.Forms.ComboBox) sender;

        comboBox.DroppedDown = true;
    }

    private void ComboBox_MouseLeave(object sender, EventArgs e)
    {
        System.Windows.Forms.ComboBox comboBox =
(System.Windows.Forms.ComboBox) sender;

        comboBox.DroppedDown = false;
    }

    private void Advenced_search_CheckedChanged(object sender, EventArgs
e)
    {
        movie_combo_box.Visible = !movie_combo_box.Visible;
    }

```

```

serial_combo_box.Visible = !serial_combo_box.Visible;
cartoons_combo_box.Visible = !cartoons_combo_box.Visible;

if (cartoons_combo_box.Visible == false)
{
    horizon_border.Location = new System.Drawing.Point(-26, 50);
    ScrollBarr.Location = new System.Drawing.Point(main_bg.Width
- 15, 52);
    ScrollBarr.Size = new System.Drawing.Size(15, main_bg.Height
- 100);
    Video_Panel.Location = new System.Drawing.Point(0, 52);
    Video_Panel.Size = new System.Drawing.Size(main_bg.Width -
15, main_bg.Height - 100);
}
else
{
    horizon_border.Location = new System.Drawing.Point(-26, 100);
    ScrollBarr.Location = new System.Drawing.Point(main_bg.Width
- 15, 102);
    ScrollBarr.Size = new System.Drawing.Size(15, main_bg.Height
- 150);
    Video_Panel.Location = new System.Drawing.Point(0, 102);
    Video_Panel.Size = new System.Drawing.Size(main_bg.Width -
15, main_bg.Height - 50);
}
movie_combo_box.SelectedIndex = 0;
serial_combo_box.SelectedIndex = 0;
cartoons_combo_box.SelectedIndex = 0;
GET_Video();
Search_btn_Click(sender, e);
}

private void Search_box_Enter(object sender, EventArgs e)
{
    System.Windows.Forms.TextBox textBox =
(System.Windows.Forms.TextBox)sender;
    if (textBox.Text == "Пошук в програмі...")
    {

```



```

        textBox.Text = " ";
        textBox.ForeColor = SystemColors.WindowText;
    }
}

private void Search_box_Leave(object sender, EventArgs e)
{
    System.Windows.Forms.TextBox textBox =
(System.Windows.Forms.TextBox) sender;

    if (string.IsNullOrEmpty(textBox.Text))
    {
        textBox.Text = "Пошук в програмі...";
        textBox.ForeColor = SystemColors.GrayText;
    }
}

private void Disfocus(object sender, EventArgs e)
{
    ScrollBarr.Focus();
}

private void Clear_Click(object sender, EventArgs e)
{
    movie_combo_box.SelectedIndex = 0;
    serial_combo_box.SelectedIndex = 0;
    cartoons_combo_box.SelectedIndex = 0;
    search_box.Text = "Пошук в програмі...";
    Search_btn_Click(sender, e);
}

private void VScrollBar_Scroll(object sender, ScrollEventArgs e)
{
    foreach (Control control in Video_Panel.Controls)
    {
        control.Top = control.Top + e.OldValue - e.NewValue;
    }
    Video_Panel.PerformLayout();
}

```

```

    }

    private void Main_Window_SizeChanged(object sender, EventArgs e)
    {
        GET_Video();
    }

    private void Submit_arend(object sender, EventArgs e)
    {
        User user = new User();
        user.Suvmit_Video();
        rented_video.Text = string.Empty;
        Rented = false;
        GET_Video();
    }
}

} using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace Video_library.Models
{
    internal class Video
    {
        public string Name { get; set; }
        public string Description { get; set; }
        public int Year { get; set; }
        public string Genre { get; set; }
        public double Rating { get; set; }
        public string Country { get; set; }
        public string[] Types { get; set; }
        public int ID { get; set; }
    }
}

```

```

internal class Videos
{
    public Video[] GetVideos()
    {
        string[,] arr = My_Init();

        Video[] videos = new Video[arr.GetLength(0)];

        for (int i = 0; i < arr.GetLength(0); i++)
        {
            Video video = new Video
            {
                Name = arr[i, 0],
                Description = arr[i, 1],
                Year = int.Parse(arr[i, 2]),
                Genre = arr[i, 3],
                Rating = double.Parse(arr[i, 4]),
                Country = arr[i, 5],
                ID = i + 1
            };

            string[] types = new string[arr.GetLength(1) - 6];
            for (int j = 6; j < arr.GetLength(1); j++)
            {
                types[j - 6] = arr[i, j];
            }

            video.Types = types;

            videos[i] = video;
        }

        return videos;
    }
}

```

```

public string[,] My_Init()
{
    string filePath = "A:\\Проекты
C#\\Cursova\\Video_library\\Video_library\\movies.txt";

    string text = File.ReadAllText(filePath);

    string[] result = text.Split('{', '}');

    result = result.Where(p =>
!string.IsNullOrEmpty(p)).ToArray();

    List<string[]> nestedArray = new List<string[]>();

    foreach (string part in result)
    {
        string[] components = part.Split(';');

        components = components.Select(c => c.Trim()).Where(c =>
!string.IsNullOrEmpty(c)).ToArray();

        nestedArray.Add(components);
    }

    string[,] array = new string[nestedArray.Count,
nestedArray[0].Length];
    for (int i = 0; i < nestedArray.Count; i++)
    {
        for (int j = 0; j < nestedArray[i].Length; j++)
        {
            array[i, j] = nestedArray[i][j];
        }
    }

    return array;
}
}

```

```

using Microsoft.VisualBasic.ApplicationServices;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Video_library.Models
{
    internal class User
    {
        public int Id { get; set; }
        public string Addres { get; set; }
        public int User_video { get; set; }
        public bool Videos_enabel { get; set; }

        public User()
        {
            string filePath = "A:\\Проекти
C#\\Cursova\\Video_library\\Video_library\\User.txt";

            string text = File.ReadAllText(filePath);

            string[] result = text.Split(';');
            result = result.Where(p =>
!string.IsNullOrEmpty(p)).ToArray();

            if (result.Length == 0)
            {
                Random random = new Random();
                int randomNumber = random.Next(1, 1000000);

                Addres = Microsoft.VisualBasic.Interaction.InputBox("Введіть
вашу адресу, або у вас не буде можливості арендувати відео чи фільми:",
"Адреса", "null");

                Id = randomNumber;
                if (Addres == "null" || Addres == "")

```

```

        {
            MessageBox.Show("Без введенного адресу не можна буде
арендувати відео");
            Videos_enabel = false;
            File.WriteAllText(filePath, $"{Id};null;");
        }
        /*if (Addres != "null" ) {
            Videos_enabel = true;
            File.WriteAllText(filePath, $"{Id};{Addres};");
        }

        User_video = -1;*/

    }
    else
    {
        Id = int.Parse(result[0]);
        if (result[1] == "null")
        {
            if (Addres == "null")
            {
                MessageBox.Show("Без введенного адресу не можна буде
арендувати відео");
                Videos_enabel = false;
                File.WriteAllText(filePath, $"{Id};{Addres};");
            }
            else
            {
                Videos_enabel = true;
                Addres = result[1];
                File.WriteAllText(filePath, $"{Id};{Addres};");
            }
        }
    }
    else
    {

```

```

        Videos_enabel = true;
        Addres = result[1];
        File.WriteAllText(filePath, $"{Id};{Addres}");

    }

    if (result.Length <= 2 || result.Length < 2)
    {
        User_video = -1;

    }
    else
    {
        User_video = int.Parse(result[2]);

    }

}

}

public void Add_Video(int video_id, string Text)
{
    string filePath = "A:\\Проекты
C#\\Cursova\\Video_library\\Video_library\\User.txt";
    User_video = video_id;

    File.WriteAllText(filePath, $"{Id};{Addres};{User_video}");
}

public void Suvmit_Video()
{
    string filePath = "A:\\Проекты
C#\\Cursova\\Video_library\\Video_library\\User.txt";
    User_video = -1;
    File.WriteAllText(filePath, $"{Id};{Addres};{User_video}");
}

}}

```