**ֿ**

**המכללה האקדמית תל חי – החוג למדעי המחשב**

**סמסטר אביב – תשפ״ג - 2023**

דו״ח סיכום פרויקט פרטני

נושא: אפליקציית מחשב לבסיס נתונים של סרטים

סטודנט מבצע: ים אלגבסי 208698266

A screenshot of a computer screen

Description automatically generated with low confidenceמרצה מנחה: איעד סולימן Phd.

# תוכן עניינים:

|  |  |
| --- | --- |
| שער | 1 |
| תוכן עניינים | 2 |
| פרק 1: מבוא | 3 |
| פרק 2: תיאור כללי | 4 |
| פרק 3: מודלים | 5-6 |
| פרק 4: תיאור מפורט של מערכת ה-API | 7-8 |
| פרק 5: תיאור מפורט של מערכת ה-WPF | 9-11 |
| פרק 6: סיכום | 12 |

פרק 1: מבוא

# תקציר:

רעיון הפרויקט הוא לקיחת תחביב – סרטים ולשלב אותו עם עבודת תכנות המשלבת עבודה עם נושאים שנלמדו במהלך התואר. הקורסים שבעיקר התבססו עליהם עבודת הגמר הם ״״סדנת תוכנה בשפת C# וסביבת .NET״ עם מר גדי שור כאשר חומר הלימוד החופף הוא בעיקר השפה בא נכתב הפרויקט ושל סביבת העבודה עם דגש על עולם ה-Full stuck, קורס ״מערכות מסדי נתונים״ לניהול בסיס הנתונים. ארגון המודלים הקיימים במודל ועבודה איתם.

# טכנולוגיות:

שפה – C#.

סביבת עבודה – .NET 7.0 Visual Studio 2023

חבילות - Entity Framework packages: CORE, TOOLS, SQL SERVER

# מבנה הפרויקט:

הפרויקט מחולק לשתי מערכות:

* אפליקציית API – הורדת נתונים ומידע באמצעות שרת אינטרנט המבוסס פרוטוקול HTTP.
* אפליקציית WPF – אפליקציה לצד המשתמש בעלת מסכים וממשקיות עם הנתונים.

פרק 2: תיאור כללי

# מודלים:

1. חברה (Company) - חברות שיוצרות סרטים. (לדוגמה: דיסני, מארוול, פיקסאר...)
2. כוכב (Stars) – כוכב, דומות בעלת תפקיד בסרט, יכול להיות שחקן בסרט ויכול להיות גם המפיק של הסרט.
3. סרט (Movie) - יצירת קולנוע על ידי חברה, כוכב קולנוע מפיק אותו ומספר מפיקים יכולים לשחק בו.
4. משתמש (User) – כל משתמש יש שם משתמש ייחודי לו. יש שני סוגים של משתמשים – מנהל ורגיל.
5. כוכב קולנוע (MovieStar) מודל קישור כדי הנועד לקשר בין הסרטים לכוכבים המשחקים בהם.

מערכת ה-API משתמשת בבסיס נתונים מסוג SQL ומכילה טבלה לכל מודל, לכל מודל קיים Contraller לגישה אל הנתונים על מנת פעולות הוספה, עריכה ומחיקה.

מערכת ה-WPF, מציגה מסכים לכל מודל ולממשק המשתמש, למערכת זו קיים מודל נוסף UserService, מודל המממש תבנית תכן מסוג סינגלטון בעל פונקציות של גישה לשרת והבאה של נתונים ופונקציות סטטיות המשומשות בפרויקט באופן מרובה.

שמירת תמונות – במהלך הפרויקט נשמר לא מעט תמונות בבסיס הנתונים כמחרוזת, זה לאחר קידוד של התמונה לצורת 64 ביט.

# תרשים זרימה של משתמש:

פרק 3: מודלים

[Key]

public string Name { get; set; }

public string Logo { get; set; }

public DateTime Founded { get; set; }

חברה (Company):  
לכל חברה יש שם הייחודי רק לה.  
ניתן להוסיף לוגו לחברה, אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.  
לכל חברה יש תאריך בה היא הוקמה.  
\*חברות וותיקות שרק שנת הקמתן ידועה תאריכן יהיה היום הראשון לאותה שנה

[Key]

public string Id { get; set; }

public string Name { get; set; }

public DateTime DateOfBirth { get; set; }

public string Image { get; set; }

כוכב (Stars):  
לכל כוכב יש id ייחודי לו שמוענק לו בעת יצירת האובייקט.  
לכל כוכב יש שם מלא ותאריך בו הוא נולד.  
ניתן להוסיף לוגו לחברה, אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.

[Key]

public string Id { get; set; }

public string Title { get; set; }

public string Poster { get; set; }

public string Director\_Id { get; set; }

public string Company { get; set; }

public DateTime ReleaseDate { get; set; }

סרט (Movie):  
לכל סרט יש id ייחודי לו שמוענק לו בעת יצירת האובייקט.  
לכל כוכב יש שם - כותר ותאריך בו הוא יצא להקרנים.  
ניתן להוסיף פוסטר לסרט, אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.  
ניתן להוסיף מפיק לסרט, מפתח זר למודל Star אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.  
ניתן להוסיף חברה לסרט, מפתח זר למודל Company אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.

משתמש (User):   
לכל משתמש יש שם משתמש ייחודי לו שמוענקת לו בעת יצירת האובייקט. (שמות אלה מאוחסנים באותיות ריקות).  
לכל משתמש יש תאריך לידה, שם פרטי ושם משפחה.  
לכל משתמש יש סיסמה מתאימה.  
ניתן להוסיף למשתמש תמונה/מייל, אם לא קיים ערך זה יהיה שווה למחרוזת הריקה.  
משתמש יכול להיות משתמש רגיל או מנהל.

[Key]

public string Username {

get { return username; }

set { username = value.ToLower(); }

}

public string FirstName { get; set; }

public string LastName { get; set; }

public string Image { get; set; }

public string Email { get; set; }

public string Password { get; set; }

public DateTime DateOfBirth { get; set; }

public bool IsAdmin { get; set; }

כוכב קולנוע (MovieStar):  
מודל קישור בין סרטים לכוכבים המייצג את השחקנים של כל סרט, לכל סרט יכול להיות כמה כוכבים שמשחקים בו וכל כוכב יכול לשחק בכמה סרטים.  
לכל אובייקט יש id ייחודי לו שמוענק לו בעת יצירתו. ואת המפתחות הזרים של הסרט והכוכב.

[Key]

public string MovieStarId { get; set; }

public string StarId { get; set; }

public string MovieId { get; set; }

שירות משתמש (UserService):  
מודל המממש תבנית תכן מסוג סינגלטון בעל פונקציות של גישה לשרת והבאה של נתונים ופונקציות סטטיות המשומשות בפרויקט באופן מרובה.  
מכיל אובייקט מסוג HttpClient לבקשות HTTP ו- baseUrl המייצג את הכתובת הבסיסית של שרת ה-api.  
מודל זה בשימוש רק במערכת ה-WPF.

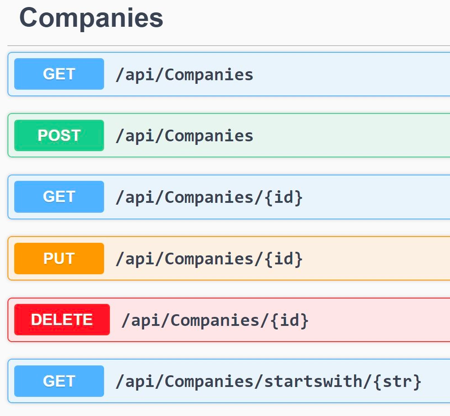
private static readonly UserService instance = new UserService();

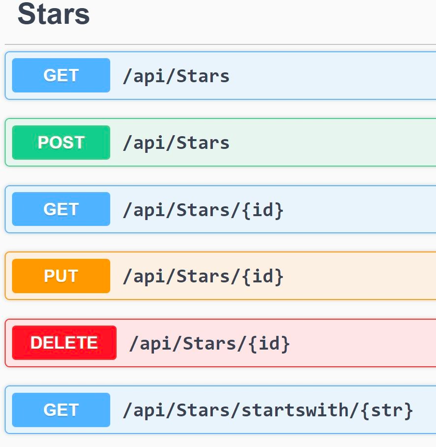
private static readonly HttpClient httpClient;

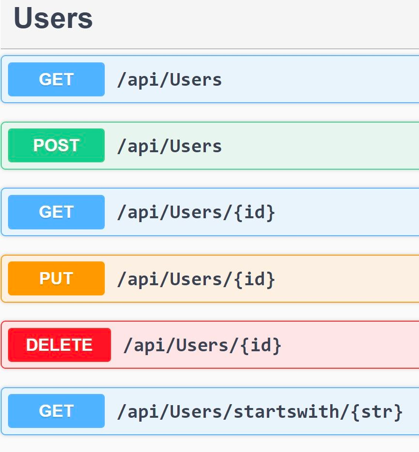
private static readonly string baseUrl = "https://localhost:7206";

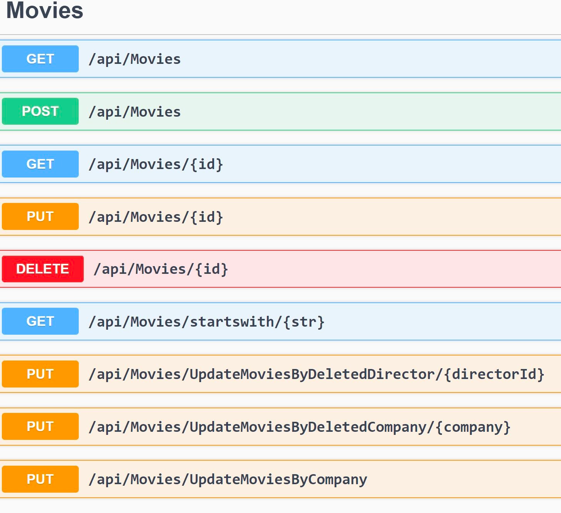
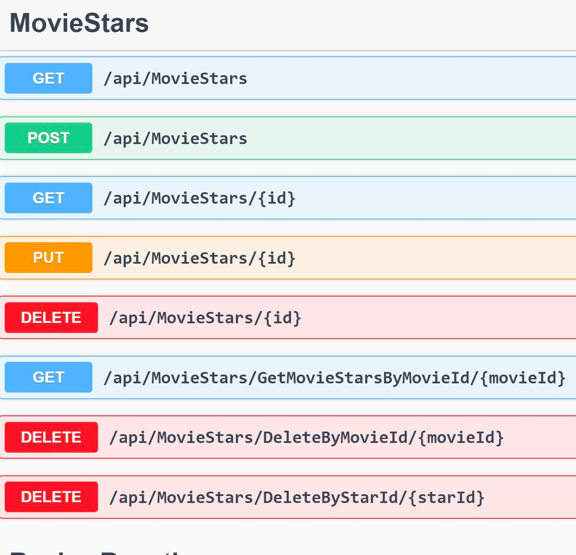
פרק 4: תיאור מפורט של מערכת ה-API

# הסבר: החלק של צד השרת בפרויקט (Server Side) המודל שאחראי לשמירת הנתונים (מודל בשם AppDbContext) משתמש במודלים המתוארים בפרק 3 לשמירת הנתונים בקובץ SQL במערכת ממנה השרת רצה. לכל מודל קיים טבלה בבסיס הנתונים ובקר (Controller) שאחראי על בקשות ה-HTTP.

בקר החברות:  
בקשות GET: 1. קבלת חברה ספציפית לפי שם החברה.  
 2. קבלת רשימת חברות מלאה.  
 3. קבלת רשימה של חברות לפי תחילית של השמות שלהן.  
בקשת POST: שמירת חברה בבסיס נתונים.   
בקשת PUT: קבלת חברה ועדכון שלה לפי השם שמזהה אותה.  
בקשת DELETE: מחיקת חברה מבסיס הנתונים לפי השם שלה.

בקר הכוכבים:  
בקשות GET: 1. קבלת כוכב ספציפי לפי מחרוזת הזיהוי שלו.  
 2. קבלת רשימת כוכבים מלאה.  
 3. קבלת רשימה של כוכבים לפי תחילית של השמות שלהן.  
בקשת POST: שמירת כוכב בבסיס נתונים.   
בקשת PUT: קבלת כוכב ועדכון שלו לפי מחרוזת הזיהוי שלו.  
בקשת DELETE: מחיקת כוכב מבסיס הנתונים לפי מחרוזת הזיהוי שלו.

בקר המשתמשים:  
בקשות GET: 1. קבלת משתמש ספציפי לפי שם המשתמש המזהה אותו.  
 2. קבלת רשימת משתמשים מלאה.  
 3. קבלת רשימה של משתמשים לפי תחילית של השמות משתמשים שלהם.  
בקשת POST: שמירת משתמש בבסיס נתונים.   
בקשת PUT: קבלת משתמש ועדכון שלו לפי שם המשתמש שלו.  
בקשת DELETE: מחיקת משתמש מבסיס הנתונים לפי שם המשתמש שלו.

בקר הסרטים:  
בקשות GET: 1. קבלת סרט ספציפי לפי המחרוזת המזהה שלו.  
 2. קבלת רשימת סרטים מלאה.  
 3. קבלת רשימה של סרטים לפי תחילית של השמות שלהם.  
בקשת POST: שמירת סרט בבסיס נתונים.   
בקשת PUT: 1. קבלת משתמש ועדכון שלו לפי שם המשתמש שלו.  
 2. לפי מחרוזת הזיהוי של כוכב, הורדה שלו מכל הסרטים שהפיק.  
 3. לפי שם של חברה, הורדה שלה מכל הסרטים שיצרה.  
 4. קבלת משתנה שמכיל ששם ישן של חברה ושם חדש  
 ושינוי של כל הסרטים עם השם הישן לשם החדש.  
בקשת DELETE: מחיקת סרט מבסיס הנתונים לפי השם שלה.

בקר הכוכבי קולנוע:  
בקשות GET: 1. קבלת שחקן ספציפי לפי המחרוזת שמזהה אותו.  
 2. קבלת רשימת שחקנים מלאה.  
 3. קבלת רשימה של כוכבים לפי מחרוזת הזיהוי של הסרט.  
בקשת POST: שמירת שחקן בבסיס נתונים.   
בקשת PUT: קבלת שחקן ועדכון שלו לפי מחרוזת הזיהוי שלו.  
בקשת DELETE: 1. מחיקת שחקן מבסיס הנתונים לפי מחרוזת הזיהוי שלו.  
 2. מחיקת כל השחקנים ששיחקו בסרט לפי מחרוזת הזיהוי שלו.  
 3. מחיקת כל השחקנים לפי מחרוזת הזיהוי של הכוכב.

פרק 5: תיאור מפורט של מערכת ה-WPF  
הסבר:  
החלק של צד הלקוח בפרויקט (Client Side) מכיל מסכים בשביל ממשקיות עם המשתמש ומשתמש במודלים המתוארים בפרק 3. כל מסך משתמש במודל UserServcie לשם שימוש בבקרים שתואר בפרק 4 לניהול בסיס הנתונים לפי בחירת המשתמש.

A screenshot of a computer screen

Description automatically generated with low confidenceLogin Window:  
מסך ההתחברות של משתמשים, אם המערכת מוצאת משתמש עם שם משתמש מתאים וסיסמה נכונה המשתמש יחובר למערכת ויעבור למסך הראשי, אחרת תופיע הודעה שגיאה מתאימה.   
אם למשתמש אין חשבון במערכת הוא יכול להירשם אליה.

A picture containing screenshot, text, art

Description automatically generated

Register Window:  
מסך ההרשמה של משתמשים, אם המערכת מוצאת משתמש עם שם משתמש דומה תופיעה הודעת שגיאה, המערכת בודקת שכל השדות מלאים ותקינים, אם כן בעת לחיצת הכפתור ייוצר המשתמש אחרת תופיעה הודעת שגיאה מתאימה.   
אם למשתמש אין חשבון במערכת הוא יכול להירשם אליה.

A picture containing screenshot, cartoon, violet, art

Description automatically generated

Main Window:  
אחרי ההתחברות המשתמש מגיע למסך הראשי של התוכנית בוא הוא יכול לבחור במה הוא רוצה לצפות.  
כפתור הצפייה במשתמשים זמין אך למנהלים ומוסתר למשתמשים רגילים.

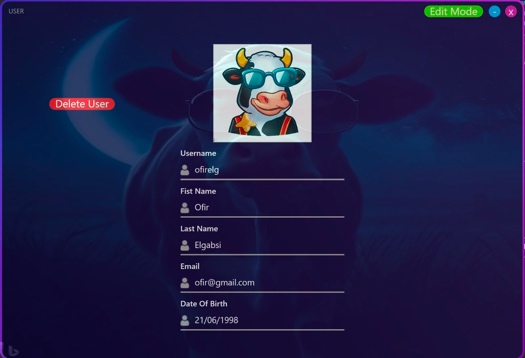
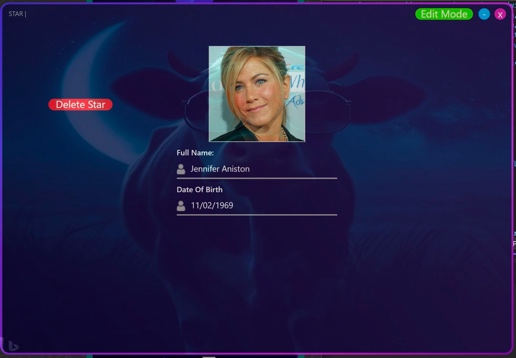
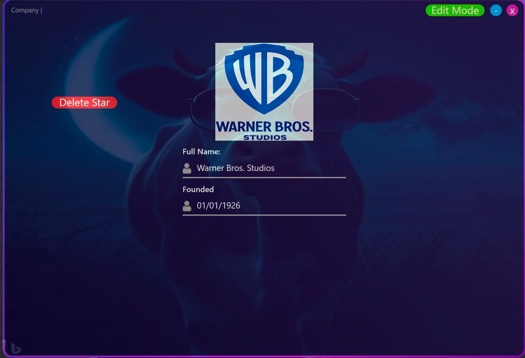
A screenshot of a computer

Description automatically generatedSearch Window:  
אחרי בעת לחיצה על אחד הכפתורים המובלים למסך זה (כולם למעת MyProfile) המשתמש יגיע לחלון החיפוש. חלון זה הוא אבסטרקטי לכל סוגי החיפוש לפי הכפתור ממנו הוא נשלח.  
המשתמש יכניס. לתיבת הטקסט המתאימה והמערכת תחפש אובייקט מתאים שמילת החיפוש הוא תחילית של שמו.  
כפתור היצירה זמין רק למנהלים ומוסתר למשתמשים רגילים.

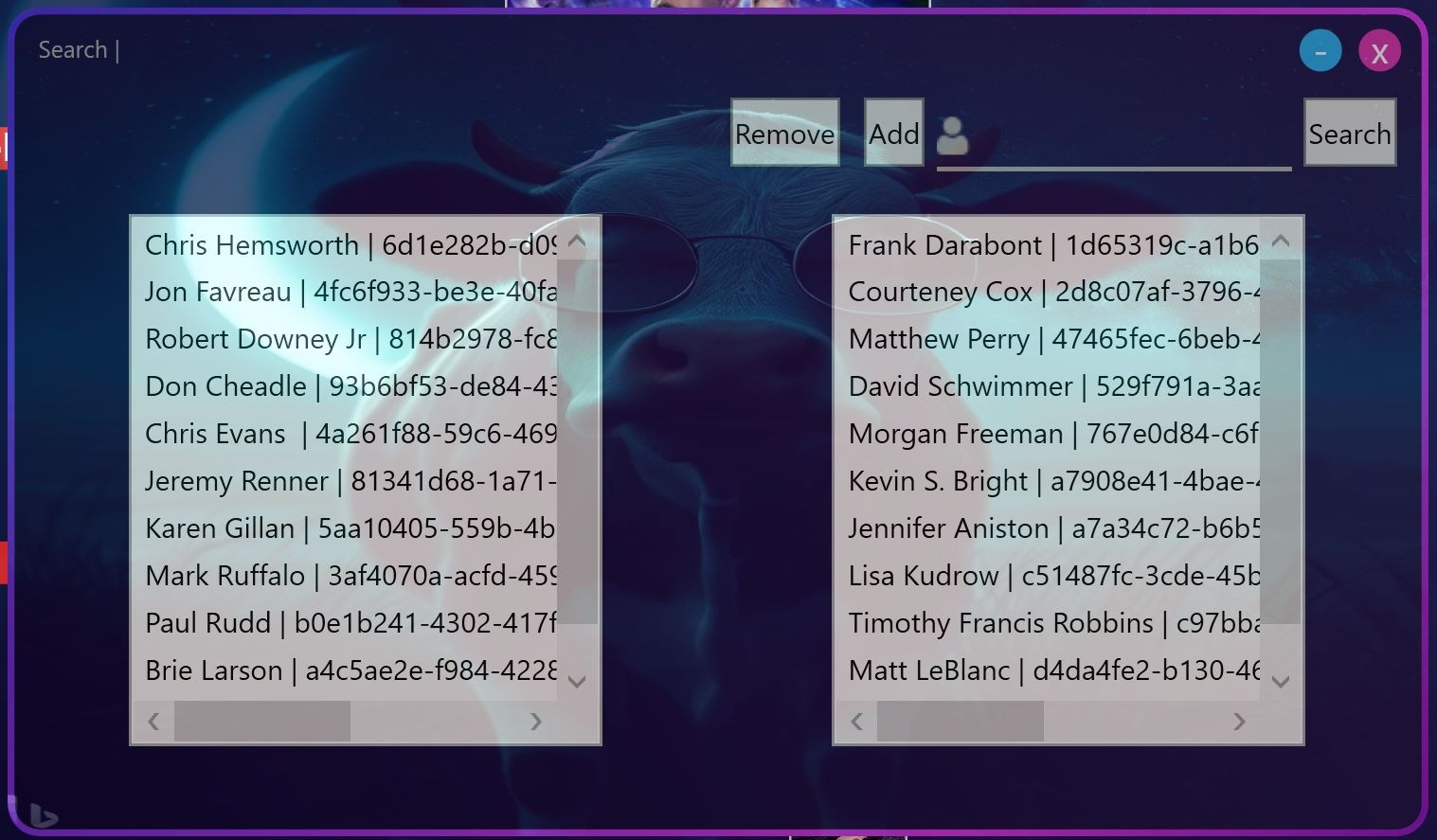
A screenshot of a movie

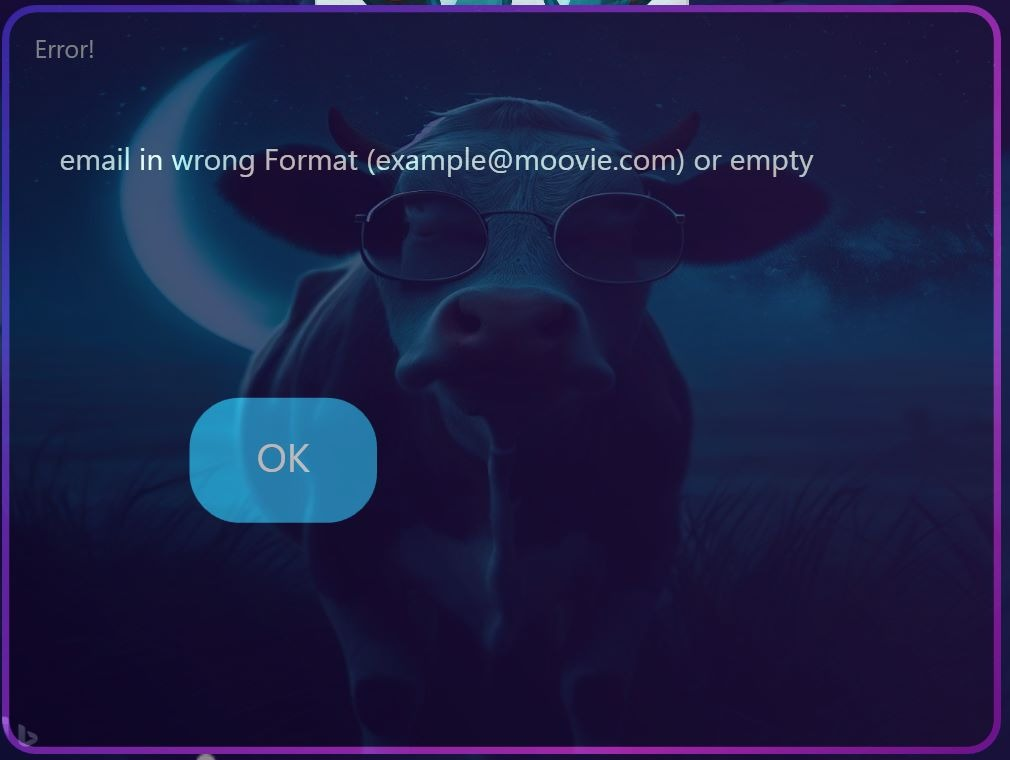
Description automatically generatedWindow Of Models:  
לכל מודל שהוא אישות יש חלון משל עצמו. כל חלון מציג את כלל הנתונים של אותו אובייקט.

* כפתור העריכה זמין רק למנהלים ומוסתר למשתמשים רגילים, כאשר נלחץ ניתן לכתוב על אותם נתונים ולשמור על כפתור השמירה שמופיעה באותו מקום של כפתור העריכה בעת הלחיצה.
* כפתור המחיקה זמין רק למנהלים ומוסתר למשתמשים רגילים. בחלון של המשתמש כפתור המחיקה לא יופיע למנהל על הפרופיל שלו ( משתמש לא יכול למחוק את המשתמש שלו).



Search Stars Mini-Window  
במצב עריכה לחיצה על כפתור השחקנים, או על תמונת המפיק ייפתח החלון לבחירת כוכבים, חלון זה מקבל כפרמטר הגבלה על הבחירה. ההגבלה היא שיש עד 200 שחקנים ומפיק אחד בסרט.



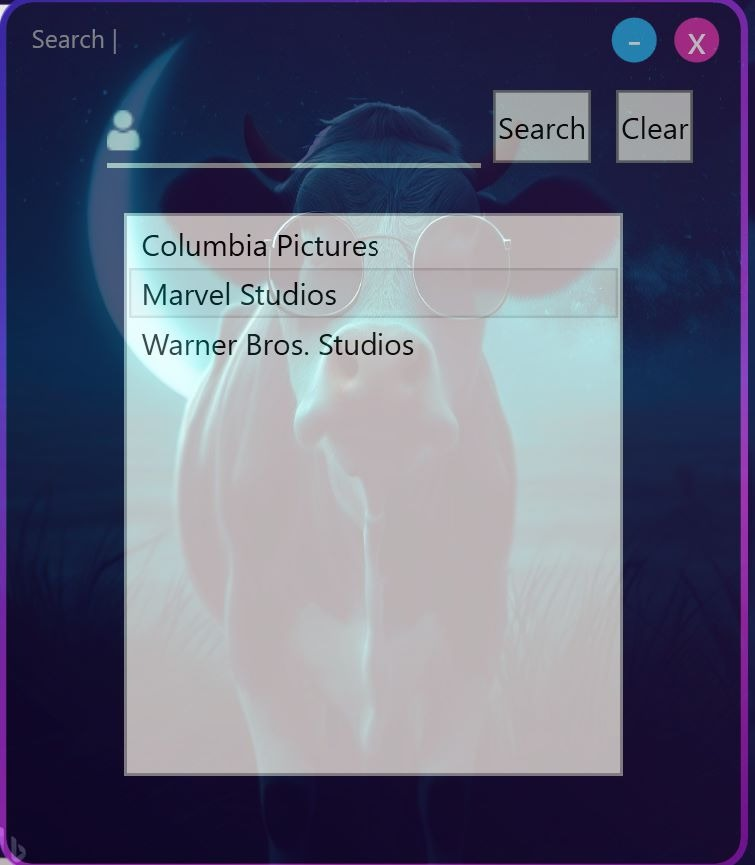
My Message Mini-Window  
חלון הודעה למשתמש ש-״מחזיר״ משתנה בוליאני.   
מקבל כפרמטרים:  
 כותרת החלון   
גוף ההודעה   
תוכן ל-True   
אפשרות - תוכן ל-False (כאשר אין הכפתור מוסתר)  


Actors Mini-Window  
בחלון הסרטים העת לחיצה על כפתור השחקנים ייפתח החלון עם רשימת השחקנים.

A screenshot of a phone

Description automatically generated with medium confidence

Search Company Mini-Window  
במצב עריכה בחלון הסרטים, לחיצה על תמונת החברה מביא לחלון ש-״מחזיר״ את החברה הנלחצת.



פרק 6: סיכום

# ביבליוגרפיה:

[אתר מייקרוסופט](https://learn.microsoft.com/en-us/dotnet/desktop/wpf/get-started/create-app-visual-studio?view=netdesktop-7.0) – מדריך ועזרה לעבודה עם Visual Studio WPF

[אתר מייקרוסופט](https://learn.microsoft.com/en-us/dotnet/desktop/wpf/get-started/create-app-visual-studio?view=netdesktop-7.0) – מדריך ועזרה לעבודה עם Visual Studio Web API With ASP.NET Core

[מודל תל חי](https://moodle.telhai.ac.il/course/view.php?id=24023) – אתר הקרוס - ״סדנת תוכנה בשפת C# וסביבת NET״ – שקפים וקבצי עזר על ידי מר גדי שור

[וויקיפדיה](https://he.wikipedia.org/wiki/%D7%A2%D7%9E%D7%95%D7%93_%D7%A8%D7%90%D7%A9%D7%99) – לנתונים ההוכנסו לבסיס הנתונים.

[Microsoft Bing Image Creator](https://www.bing.com/create) - מחולל תמונות באמצעות בינה מלאכותית של מיקרוסופט לתמונות המערכת.

[RJ Code Advance](https://rjcodeadvance.com/) – לפיתוח ה-UI ונראות ממשק האפליקציה.

# מטרות המטרה העיקרית של הפרויקט היא העמקה בסביבת העבודה והתמקצעות בה ולבכיר ולרכוש כלים נוספים בעת השימוש בה. מטרות נוספות שהושגו במהלך הפרויקט היא חיזוק והכירות נוספת עם השפה C# ושיפור יכולת הלמידה העצמאית.

# מסקנות המסקנה החשבוה ביותר אחרי עשיית הפרויקט היא שהתכנון שקורה לפני הכתיבה עצמה הוא חשוב לא פחות, צריך לקחת את השלב הזה בחשיבות רבה כי הוא בונה ומביא ביסוס לכל שלב אחר בתהליך הפרויקט. בדיעבד לתת חשיבות יותר גדולה ל-API כדי לחסוך עבודה לתוכנה וברשימת הקוד ב-WPF.

# אפשרויות להמשך הפרויקט מכיל מידע בסיסי של סרטים וניתן יהיה להרחיב את הנתונים כמו למשל: הוספת אופציה של סדרות ולא רק סרטים, זמן סרט, במאיים, מפיקים (לא רק אחד), ז׳אנר, שפה, שם התפקיד של שחקן בסרט ועוד מגוון גדול של מידע.

הפרויקט יכול להיות בסיס לפרויקט רחב יותר כמו לדוגמה: אפליקציה לביקורות של סרטים, אפליקציה לשם צפייה בסרטים, אפליקציית חנות סרטים ואפליקציה לבית קולנוע והזמנת כרטיסים לסרט.

הרחבת הפלטפורמות – הפרויקט הוא אפליקציה למחשבים עם מערכת הפעלה Windows ולכן ניתן יהיה להרחיב ולבנות מערכות המשתמשות ב-API הנבנה במגוון רחב יותר של פלטפורמות כמו Web, אנדרואיד, IOS וכדומה.