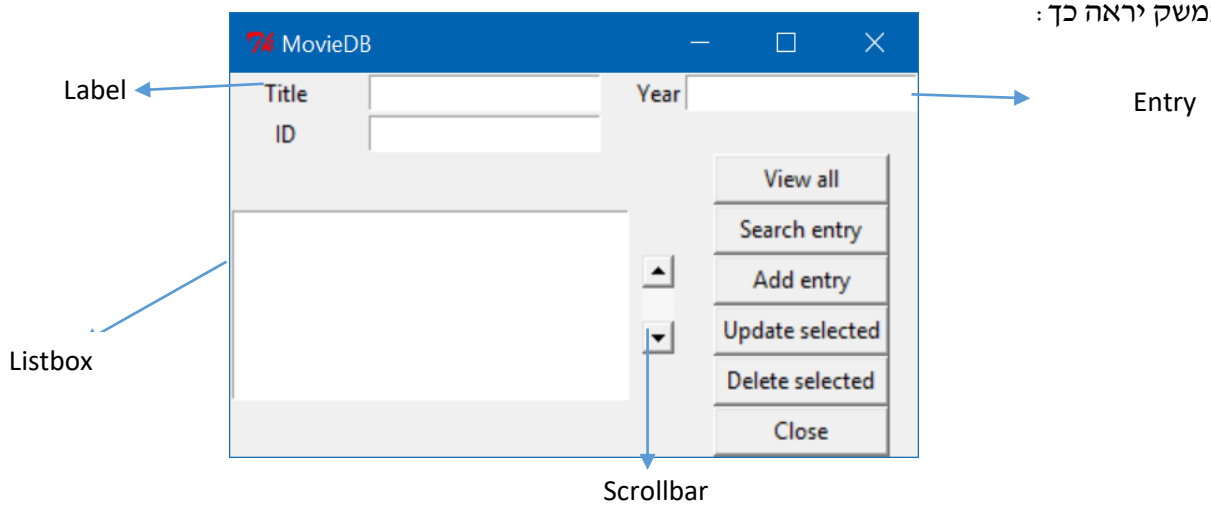
מטלת מעבדה 1

במטלה זו עליכם לבנות אפליקציה לניהול מאגר נתונים של סרטים.

הממשק יראה כך:



חלק א'

יש לממש את האפליקציה בשתי שכבות:

1. frontend.py – קוד ה-GUI (שם נעשה import לbackend)

2. backend.py – הקוד שמתממשק עם ה-DB, אותו יש ליצור באמצעות SQLITE כפי שנלמד בכיתה.

בתחילת הריצה, האפליקציה תבדוק אם קובץ ה-DB קיים, אם לא יש ליצור אותו ולמלא אותו בנתונים מהקובץ movies.csv המצורף. הנתונים בקובץ רשומים בפורמט הבא: ID,Title,Genre.

<u>הערה</u>: לחלק מהסרטים יש רשימה של Genres המופרדת ע"י '|'. יש לקחת את הראשון מביניהם ל-DB.

ביצירת הטבלה יש להגדיר את ID כ-PRIMARY KEY.

בחלק זה יש לממש את הפונקציונאליות הבאה:

1. ניתן לצפות/לטעון את כל תכולת ה-DB ב-Listbox באמצעות לחיצה על הכפתור View all.
2. ניתן לחפש סרט באמצעות הזנת נתונים בכל אחד משדות ה-GUI (או בכולם) ולחיצה באמצעות כפתור Search entry. (ניתן למימוש בשאילתה אחת, חישבו איך) לדוגמא, הכנסת השנה 2004 ולחיצה על Search entry תציג את כל הסרטים מ-2004.
3. ניתן להוסיף סרט למאגר ע"י מילוי כל השדות ולחיצה על Add entry.
4. ניתן לעדכן שדות של סרט מסוים ע"י סימונו ב-Listbox, שינוי הערכים בשדות הרלוונטים, ולחיצה על Update selected.
5. ניתן למחוק סרט היה סימונו ב-Listbox ולחיצה על Delete selected.
6. ניתן לצאת מהאפליקציה ע"י לחיצה על הלחצן Close.

חלק ב' – מימוש ראשוני של מערכת המלצה

בחלק זה תממשו web service של מערכת המלצה.

לשם מימוש מערכת המלצה תיעזרו בקובץ ratings.csv בו נמצאים דירוגי הסרטים של כל המשתמשים.

השירות יאפשר גישה לURL '/rec' גם בשיטת POST וגם בשיטת GET. כאשר הפרמטרים שישלחו לכתובת הם ה-userid שמבקש לקבל המלצות, ו-K מספר ההמלצות שהמשתמש מעוניין לקבל.

(בגוף הפוסט פרמטרים אלה יהיו תחת השדות userid ו-k בהתאמה)

דוגמא לבקשת GET:

http://127.0.0.1:5000/?userid=1234&k=6

דוגמא לבקשת POST:

```
requests.post("http://127.0.0.1:5000", data={'k': 5, 'userid': 1234})
```

השיטה תחזיר מבנה מסוג JSON אשר יכיל את שמות K הסרטים המומלצים.

ההמלצה תתבצע על סמך מימוש שיטת collaborative filtering אשר מבוססת על דירוגים זהים בין משתמשים ובמציאת הדמיון ע"י Pearson correlation.

$$userSim(u,n) = \frac{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)(r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \subset CR_{u,n}} (r_{ni} - \bar{r}_n)^2}}$$

בנוסחה זו, מודדים את הדמיון בין user u לבין user n, כאשר משווים את כל הדירוגים המשותפים למשתמשים אלו.

$CR_{u,n}$ מתייחס לסט הדירוגים של הסרטים המשותפים בין המשתמשים u ו-n.

$\bar{r}_u$ מתייחס לדירוג הממוצע של משתמש u.

עליכם למצוא את K המשתמשים הכי דומים למשתמש שהוזן, ומכל אחד מהם לקחת את הסרט שמדורג הכי גבוה ברשימה שלהם. אם הסרט הזה כבר קיים ברשימה שאספנו, יש לקחת את השני הכי גבוה וכך הלאה.

לצורך נוחות, התייחסו רק לסרטים **המשותפים** בין המשתמשים.

חשוב: גישה לDB תתבצע אך ורק מהקובץ backend.py.

דגשים

1. יש לשמור על תקינות הקלט ולהציג הודעות שגיאה מתאימות.

2. יש לתעד את הקוד.

3. יינתן ציון גבוה יותר לקוד מסודר יותר. יש לשמור על חלוקה נכונה לפונקציות ועל מודולאריות האפליקציה.

הגשה:

- יש להגיש את המטלה עד לתאריך 22.4.18.
- ההגשה תתבצע ע"י קובץ zip המכיל את הקוד שנכתב בהתאם לדרישות. פורמט קובץ ה-zip יהיה ID01_ID02.zip.
- ההגשה הינה **בזוגות**.
- חבר קבוצה אחד בלבד יעלה את הפתרון לאתר.
- בעיות אישיות בנוגע למועד ההגשה יש להפנות לבודק התרגילים הקורס טרם מועד ההגשה.
- **כל חריגה מנהלים אלו, ללא אישור בכתב מצוות הקורס, מהווה עילה לפסילת המטלה או להפחתת נקודות.**
- **אין להעתיק פתרונות ואין לשתף קוד בין סטודנטים. אין להעתיק קוד מוכן באינטרנט!**
- **לפתרון המטלה יש להשתמש בגרסת פייטון 2.7 בלבד ובחומר הנלמד במסגרת ההרצאות בקורס בלבד.**
- להבהרות, הכוונות או כל עזרה אחרת יש לפנות לפורום המטלות במודל.
- בדיקת המטלה תתייחס בין השאר לפרמטרים הבאים: נכונות הקוד, יעילות הקוד וזמני ריצה. יש לבדוק מקרי קצה.

בהצלחה!