

## פרויקט קורס חלק ב' – DJANGO

### הנחיות להגשת התרגיל:

1. **תאריך הגשה - 25/08/24 בשעה 23:55.**
2. **מתרגלת אחראית על התרגיל – קורל שרף.**
3. יש להגיש את תרגיל הבית בזוגות בלבד. ציון המטלה של סטודנט שיגיש לבד ללא אישור יתחיל מ-90.
4. יש להגיש קובץ zip יחיד המכיל את קבצי הקוד המשמשים לבניית האתר בהתאם להנחיות ההגשה הרשומות בהמשך המסמך. פורמט שם קובץ ה-zip הינו : id1\_id2.zip, כאשר id1 ו-id2 הם מספרי הזהות של המגשים.
5. עליכם להשתמש לצורך הפתרון באפליקציה יחידה ששמה Contents\_App (לשים לב לכך ש-C ו-A נכתבות כ-capital letters).
6. את הקובץ יש להגיש דרך אתר הקורס במקום המתאים ב-moodle על ידי **אחד** מבני הזוג.
7. איחור בהגשת התרגיל יגרור קנס בגובה 20% מהציון עבור כל יום איחור (פרט למקרים חריגים כגון מילואים. במקרים אלה יש לפנות למתרגל טרם הגשת התרגיל).
8. 5 נק' מהציון על התרגיל יינתנו על מילוי הנחיות 3-6 המופיעות לעיל, ועל ההנחיות המפורטות בסוף מסמך זה.

## הקדמה

בחלק זה של הפרויקט עליכם להקים אתר למערכת המבוססת על החלק הראשון של הפרויקט, בהתאם להוראות המפורטות בעמודים הבאים.

### **עליכם לבנות את המערכת באמצעות אפליקציה יחידה ששמה `Contents_App`.**

אין להשתמש בשם אפליקציה אחר! שימו לב לכך ש-C ו-A נכתבות כ-capital letters.

החיבור למסד הנתונים יתבצע בהתאם להנחיות המופיעות בתרגול 8. שימו לב כי חלק משמות הרלציות/שדות עשוי להפוך ל-lowercase.

## תיאור המערכת

הרלציות בהן תשתמשו בחלק זה של הפרויקט מוגדרות בהשראת הסיפור מחלק א', אך ייתכנו מספר הבדלים. לפני תחילת העבודה על התרגיל, מחקו את כל הרלציות שיצרתם עבור חלק א' של הפרויקט. לקבצי הפרויקט מצורף קובץ בשם `create_table_commands.sql` המכיל פקודות ליצירת הטבלאות במסד. עליכם להריצו דרך ה-console לפני תחילת העבודה על האתר. יש להקפיד על סדר ההרצה כפי שמופיע בקובץ. בנוסף, במודל מופיעים קבצי CSV לדוגמה שתוכלו לטעון לרלציות. מומלץ לעבור על קבצים אלו לפני תחילת העבודה.

הרלציות בהן תשתמשו בחלק זה הן:

**Movies** (title, genre, releaseDate, budget)

**ActorsInMovies** (aName, mTitle, salary)

**Users** (uID, country, favActor)

**Watching** (uID, mTitle, wDate, rating)

עבור כל אחת מהרלציות, ה-primary key מסומן באמצעות קו תחתון.

## תיאור הרלציות:

### **Movies – רלציה הכוללת מידע אודות סרטים.**

- title – שם הסרט.
- genre – סוגת הסרט.
- releaseDate – תאריך יצירת הסרט.
- budget – תקציב הסרט בדולרים.

### **ActorsInMovies – רלציה הכוללת מידע אודות השתתפות של שחקן בסרט.**

- aName – שם השחקן.
- mTitle – שם הסרט.
- salary – השכר על השתתפות השחקן בסרט.

### **Users – רלציה הכוללת מידע אודות משתמשים.**

- uid – מזהה המשתמש.
- country – המדינה בה המשתמש גר.
- favActor – השחקן האהוב על המשתמש.

### **Watching – רלציה הכוללת תיעודי צפייה של משתמש בסרט בימים שונים.**

- uid – מזהה המשתמש.
- mTitle – שם הסרט.
- wDate – תאריך הצפייה.
- rating – הדירוג שנתן המשתמש לסרט במועד בו צפה בו (מספר שלם בין 1 ל-5).

להלן תיאור הדפים השונים אשר אמור להכיל האתר שתבנו :

## 1. דף הבית

דף הבית של האתר צריך לכלול את הפרטים הבאים :

- כותרת לאתר המערכת.
- תמונה מייצגת שמתאימה לאתר (לבחירתכם)
- קישורים לשאר העמודים אשר מתוארים בגיליון. מיקומם של הקישורים בעמוד לא קריטי כל עוד הם בולטים לעין עבור המשתמש. יש ליצור בכל עמוד (לא רק בעמוד הבית) קישור לכל אחד מהעמודים האחרים.

להלן דוגמה לעמוד :

### Welcome to the Contents Website



[Query Results](#)  
[Add Actor to Movie](#)  
[Record Watching](#)

## 2. הצגת תוצאות שאילתות

לפני תחילת העבודה על עמוד זה מומלץ לקרוא את הנספח המופיע בסוף הנחיות התרגיל. בעמוד זה יש ליצור, בנוסף לקישורים המובילים לעמודים האחרים, טבלה עבור כל אחת משלוש השאילתות הבאות, אשר בה יוצגו תוצאות השאילתה. יש להוסיף כותרת לכל אחת מטבלאות המודיעה על מספר השאילתה (ראו דוגמה בעמוד הבא).

הערה: בחלק זה אתם יכולים להשתמש בשאילתות VIEW. אם ברצונכם לעשות זאת, עליכם לכתוב אותן בקובץ SQL בשם view\_queries ולהגיש אותן יחד עם שאר קבצי התרגיל כפי שמפורט בהנחיות ההגשה שבסוף מסמך זה.

### עבור כל שאילתה מותר להשתמש ב-4 שאילתות VIEW לכל היותר.

a. הכנסה של סרט מוגדרת להיות ההפרש בין תקציב הסרט למשכורות של כל השחקנים בסרט. סרט יוגדר כ"הצלחה מסחררת" אם הוא בין 20 הסרטים בעלי ההכנסה הגבוהה ביותר מבין הסרטים במסד.

החזירו לכל ז'אנר שכולל סרט אחד לפחות שהוא הצלחה מסחררת את מספר הסרטים שמוגדרים כהצלחה מסחררת, ואת הדירוג הממוצע שצופים העניקו לסרט כזה מעוגל כ-2 ספרות אחרי הנקודה), כאשר הממוצע מחושב על פני כל הסרטים באותו הז'אנר. יש להחזיר את המידע בצורה ממוינת בסדר יורד לפי מספר הסרטים שמוגדרים כהצלחה מסחררת.

b. "סרט משתפר" הוא סרט שבו כל משתמש שצפה בו יותר מפעם אחת, העניק לסרט דירוג גבוה יותר בכל צפייה בהשוואה לדירוג שהעניק לאותו הסרט בצפיות קודמות. מבין השחקנים ששיחקו בלפחות 3 סרטים משתפרים, מצאו את השחקנים שאף משתמש לא אוהד אותם. החזירו לכל שחקן כזה את שם השחקן, ואת מספר הסרטים בו הוא משחק.

c. "אוהד מדומה" הוא משתמש אשר מקיים את שני התנאים הבאים:  
i. המשתמש לא צפה באף סרט בו השתתף השחקן המועדף עליו.  
ii. מספר הצפיות של המשתמש גדול ממש ממספר הצפיות הממוצע של משתמשים שגרים באותה מדינה.

לכל מדינה שכוללת לפחות 5 אוהדים מדומים, החזירו את מזהה האוהד המדומה בעל מספר הצפיות הגדול ביותר. במקרה שקיימים מספר אוהדים מדומים בעלי מספר הצפיות הגדול ביותר, יש להחזיר את כולם. יש להחזיר את המידע בצורה ממוינת בסדר עולה לפי שם המדינה.

# Query Results

## Query 1:

Genre	Number of Movies	Average Rating

## Query 2:

Actor Name	Number of Movies

## Query 3:

Country	User ID

- [Home](#)
- [Add Actor to Movie](#)
- [Record Watching](#)

### 3. הוספת שחקן לסרט

בעמוד זה עליכם ליצור טופס המאפשר הוספת תיעוד של השתתפות שחקן בסרט. בטופס יופיעו 3 שדות חובה שבהם המשתמש צריך להזין את שם השחקן, שם הסרט, והמשכורת שהשחקן מבקש לקבל עבור השתתפותו בסרט. עליכם לאכוף ברמת ה-HTML את חובת מילוי כל השדות.

הזנה של שחקן בסרט לא תאושר אם מתקיים אחד או יותר מהמקרים הבאים:

- הסרט לא קיים במערכת בטבלת הסרטים.
- השחקן כבר מתועד במערכת כשחקן בסרט.
- השכר שהשחקן ביקש גדול ממש מהפרש ההכנסה שמתועדת לסרט פחות המשכורות של כל השחקנים האחרים.

מתחת לשדות עליכם להציג כפתור להגשת הטופס. אם הוספת התיעוד של משחק השחקן בסרט לא אושרה יש להדפיס הודעה המתארת את הסיבה לכך. יש להדפיס הודעת שגיאה אחת, גם אם מספר השגיאות גבוה יותר. סדר העדיפות בהדפסת השגיאות הוא כסדר הופעת התנאים כפי שמופיע לעיל. אם ההוספה הצליחה יש לעדכן את הרלציה שמתעדת את השחקנים שמשחקים בסרט, ולהציג הודעת הצלחה.

כמו כן, לאחר הגשת הטופס, תופיע בעמוד טבלה שתציג את פרטי 5 הסרטים האחרונים (תאריך ההעלאה שלהם הוא המאוחר ביותר) בהם השחקן שיחק. יש להציג את הסרטים בצורה ממוינת יורדת לפי תאריך ההעלאה. במקרה שקיימים פחות מ-5 סרטים אחרונים, יש להציג רק אותם.

להלן דוגמה לעמוד (לאחר לחיצה על הכפתור submit):

### Add Actor to Movie

Actor Name:

Movie Title:

Salary:

Submit

The actor was successfully added to the movie.

### Last 5 Movies:

Title	Genre	Release Date

[Home](#)  
[Query Results](#)  
[Record Watching](#)

#### 4. טופס תיעוד צפייה בסרט

בעמוד זה יוכל המשתמש להכניס תיעוד חדש של צפייה של משתמש בסרט ביום כלשהו.

המשתמש יזין באמצעות טופס את מספר הזהות של המשתמש, שם הסרט, תאריך הצפייה והדירוג לסרט. סוגי השדות צריכים להיות תואמים לאלו שמופיעים בדוגמה שבתחתית עמוד זה.

תיעוד צפייה של משתמש בסרט לא תאושר אם מתקיים אחד או יותר מהמקרים הבאים:

- קיים כבר תיעוד של צפייה של המשתמש באותו הסרט ביום שהוזן בטופס.
- קיים למשתמש תיעוד של צפייה באותו הסרט במועד מאוחר יותר מהמועד שהוזן בטופס.
- התאריך שהוזן בטופס מוקדם מתאריך ההעלאה של הסרט שעבורו נרצה לתעד את הצפייה.

מתחת לשדות עליכם להציג כפתור להגשת הטופס. אם הוספת התיעוד לא אושרה יש להדפיס הודעה המתארת את הסיבה לכך. יש להדפיס הודעת שגיאה אחת, גם אם מספר השגיאות גבוה יותר. סדר העדיפות בהדפסת השגיאות הוא כסדר הופעת התנאים כפי שמופיע לעיל. אם ההוספה הצליחה יש לעדכן את הרלציה שמתעדת צפיות של משתמשים בסרט, ולהציג הודעת הצלחה.

להלן דוגמה לעמוד:

## Record Watching

User ID:

Movie Title:

Watching Date:

Rating: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

[Home](#)  
[Query Results](#)  
[Add Actor to Movie](#)



## הנחיות הגשה

התרגיל ייבדק מתוך מסד נתונים של הסגל. על כן אין להגיש את תיקיית הפרויקט (המכילה את הקבצים init, settings, urls ו-wsgi).

יש להגיש בתוך קובץ zip (שיקרא לפי הפורמט ID1\_ID2, ראו הנחיה 3 בעמוד 1) ובתוכו 3 תיקיות וקובץ שאלות views (אם אכן יצרתם כאלה) באופן הבא:

**ID1\_ID2**

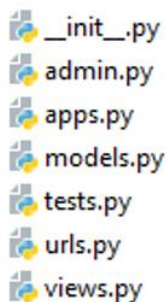
**Contents\_App**

**templates**

**static**

view\_queries.sql

כאשר Contents\_App (מזכירים שוב כי שם זה חייב להיות שם האפליקציה גם אצלכם) היא תיקיית האפליקציה שלכם אשר אמורה להכיל את הקבצים הבאים:



- \_\_init\_\_.py
- admin.py
- apps.py
- models.py
- tests.py
- urls.py
- views.py

templates היא התיקיה בה מופיעים קבצי ה-html.

static היא התיקיה בה מוגדרים הקבצים הסטטיים (תמונה, וקובץ ה-css אם אכן תבחרו ליצור). אין צורך להגיש את תיקיה זו אם לא השתמשתם בקבצים סטטיים.

view\_queries.sql הוא קובץ שאלות view (שימו לב כי מדובר ב**שאלות** view ולא בקובץ views של Django). עליכם לכתוב את שאלות ה-views בתוך הקובץ עם הפרדה של שתי שורות בין שאלתה לשאלתה, ולסמן את סופה של כל שאלתה באמצעות התו ";". לפני ה-views של שאלתה מסוימת, כתבו הערה "Views for Query <num>" כאשר <num> זה מספר השאלתה.

הערה - אתם רשאים לייצר קובץ python נוסף בתיקיית האפליקציה בשם utils. בקובץ זה אפשר לממש פונקציות עזר בהן תוכלו לעשות שימוש מ-views.py. אם אכן תבחרו באפשרות זו, לא לשכוח לכתוב

\* from utils import \*

## נספח - הרצת שאילות ב-Django

בתרגול 8 ראינו כיצד ליצור מסד נתונים וכיצד לתשאל אותו. להלן תזכורת מסודרת.

- (1) הוסיפו בראש הקובץ `views.py` (או בקובץ `utils.py` אם שם תכתבו את השאילות) את הפקודה הבאה:

```
from django.db import connection
```

- (2) צרו בקובץ `views.py` את הפונקציה הבאה:

```
def dictfetchall(cursor):  
    "Return all rows from a cursor as a dict"  
    columns = [col[0] for col in cursor.description]  
    return [dict(zip(columns, row)) for row in cursor.fetchall()]
```

- (3) כאשר תרצו להריץ שאילתה ולקבל את תוצאתה, יש לקרוא לסמן של ה-module המיובא (`cursor`) באמצעות הפקודה `with connection.cursor() as cursor`, להריץ באמצעות את השאילתה עם הפקודה `execute` ולשמור את התוצאה לרשימה (המאפשרת מעבר איטרטיבי על התוצאות) באמצעות הפונקציה `dictfetchall`.

למשל, חלק הקוד הבא ישמור לתוך המשתנה `sql_res1` את שמות כל הפוקימונים המוגדרים כאגדיים ואת ציון עצמת ההתקפה שלהם. במקרה זה, התוצאה תועבר מתוך ה-view בו מבוצעת הפעולה לקובץ `index.html` (אופן העברת התוצאה זהה לזה שראינו בתרגול):

```
with connection.cursor() as cursor:  
    cursor.execute("""  
        SELECT Name, Attack  
        FROM Pokemons  
        WHERE Legendary=1;  
    """)  
    sql_res1 = dictfetchall(cursor)  
    return render(request, 'index.html', {'sql_res1': sql_res1})
```

**בהצלחה!**