

פרויקט בסיסי נתונים – סרטים - User manual

מידע כללי וממשק משתמש

הפרויקט מאפשר לתשאל נתונים אודות כ-16,000 סרטים שנכללים במאגרי המידע של IMDB שיצאו לאקרנים בין השנים 1933 – 2020, הוא כולל 7 שאילתות SQL אשר מתשאלות מידע מ-MySQL DB ומטרתם לאפשר למשתמש גישה נוחה לבחינת פרטים ונתונים אודות הסרטים שמופיעים במאגר באמצעות ממשק אינטואיטיבי וללא כל ידע טכני.

הפרויקט בנוי מדף Index, שמשמש כ-"דף הבית" של הפרויקט המאפשר גישה מהירה לכל אחד מהדפים המציגים את השאילתות אשר מתשאלות את בסיס הנתונים לפי קלט המתקבל מהמשתמש, כל תמונה של גיבור-על משמשת לינק למעבר ישיר לשאילתה אשר פרטיה מצוינים מתחת לאותה התמונה.

שאילתות 2,4,5,6 מפנות לדפים הכוללים שדות לקליטת קלט מהמשתמש לצורך התאמת השליפה מבסיס הנתונים לצרכי המשתמש, כל אחד מדפים אלו מאפשרים חזרה לעמוד הבית באמצעות לחיצה על כפתור ה-Home Page, או לחלופין הכנסת הנתונים הרלוונטיים ולחיצה על כפתור ה-Submit לצורך מעבר לעמוד המציג את תוצאות השאילתה.

במידה והקלט אינו תקין, המשתמש מופנה לדף 404, אשר מאפשר לו לחזור אל דף הבית.

מבנה פעילות התכנית – General Flow

התכנית מבוססת על שרת flask, אשר מופעל באמצעות הקוד בקובץ app.py, השרת מגדיר את כל הנתונים לדפים המרכיבים את הפרויקט, ומאפשר לגשת לקובץ ה-index שמשמש כדף הבית. דף ה-index מאפשר מעבר לכל אחד מהשאילתות שנכתבו לצורך הפרויקט, בין אם לדף הדורש קליטת קלט מהמשתמש או לתוצאות השאילתה הרצויה וזאת כתלות בסוג השאילתה שנבחרה בדף הבית.

כאשר הקלט של השאילתה הוכנס ע"י המשתמש ונלחץ Submit הקוד בקובץ generalQueries.py נכנס לפעולה, הוא יוצר חיבור עם בסיס הנתונים הסטטי שנבנה ב-MySQL, באמצעות יצירת חיבור SSH ו-tunneling, ומעביר את השאילתה הרצויה אשר נמצאת בקובץ queries.py אל החיבור אשר מתשאל את בסיס הנתונים ומחזיר את התוצאה הרצויה.

התוצאה מוצגת בדף /q#-output אשר אליו מופנה המשתמש באופן אוטומטי.

נציין שוב כי במידה והקלט אינו תקין, המשתמש מופנה לדף 404, אשר מאפשר לו לחזור אל דף הבית.

מבנה הקוד

/DataBase Project – ספרייה המכילה את הפרויקט בכללותו, כוללת ספריית SRC המכילה את קוד הפרויקט, וספריית DOCUMENTATION אשר כוללת את תיעוד הפרויקט.

/SRC – ספרייה המכילה את קוד הפרויקט מחולקת לספריות הבאות לפי מבנה הקוד.

/APPLICATION-SOURCE CODE – ספרייה המכילה את קוד ה-Server של הפרויקט. הספרייה כוללת 3 קבצי פייתון האחראים על פעולת ה-Server של הפרויקט.

- **app.py** – שרת שמומש באמצעות Flask, השרת מגדיר את כל הנתונים לדפים הכוללים את הפרויקט, לכל שאילתה הדורשת קלט מהמשתמש קיים נתיב לדף לקליטת הקלט, לכלל השאילתות מוגדר נתיב לקובץ HTML המציג את הפלט מבסיס הנתונים.
- **queries.py** – קובץ המכיל את שאילתות ה-SQL שיתשאלו את בסיס הנתונים.
- **generateQueries.py** – יצירת חיבור לבסיס הנתונים באמצעות SSH ו-Tunneling, כולל פונקציות לשליחת השאילתות דרך החיבור לבסיס הנתונים וקבלת התוצאות לצורך הצגתם למשתמש.

/templates – ספרייה המכילה את דפי ה-HTML המוצגים למשתמש.

/static – ספרייה המכילה את קבצי CSS להוספת עיצוב לדפי התכנית, וספריית assets הכוללת תמונות בהם נעשה שימוש בקבצי ה-HTML לצרכי עיצוב בלבד.

/API-DATA-RETRIEVE – ספרייה המכילה את קובץ API-DATA-RETRIEVE.py וכוללת את קובץ הפיתון באמצעותו נאספו הנתונים מ-API TMDB אודות סרטי IMDB והוכנסו לתוך טבלאות ה-DB.

CREATE-DB-SCRIPT – יצירת ה-DB, ויצירת הטבלאות הרלוונטיות.

עבודה מול מקורות הנתונים – קובץ CSV מאתר data.world ו-API TMDB

ה-API בו נעשה שימוש הינו TMDB והנתונים ממנו נאספו בכמה שלבים שונים. ראשית מצאנו קובץ בתצורת CSV של 80 אלף רשומות של סרטים מאתר data.world אשר הכיל מידע אודות סרטי IMDB, לכל רשומה היה שדה בשם imdb_id. שלפנו מה-CSV את כל עמודת ה-imdb_id והכנסנו אותן לסט.

לאחר מכן, שלפנו סרט אחר סרט מתוך ה-API. גם כאן אחד השדות שקיבלנו הוא imdb_id. במהלך השליפה התקדמנו רק עם סרטים שנמצאו בסט שהוגדר מלעיל.

כל שליפה של סרט כללה בתוכה מספר ערכים וביניהם ערכים מצורת רשימות, כל איבר ברשימה הוא אובייקט בפני עצמו. על כן, כל ערך שתכולתו היה רשימה פוצל לטבלה משלו, לדוגמה: genres, spoken_languages וכו' כלומר: במהלך השליפה יצרנו אובייקטים של טבלאות שאותם שמרנו כ-CSV ולאחר מכן השתמשנו בסקריפט אחר שמעלה אותם ל-mysql.

ניתן לראות את ה-script בו נעשה שימוש לצורך איסוף הנתונים מתוך ה-API בפרויקט בקובץ API-DATA-RETRIEVE.py.

ספריות חיצוניות

בפרויקט נעשה שימוש בספריות פייתון הללו :

- Flask
- Mysql.connector
- sshunnel