**IMDb Project Documentation**

**By: Faisal Rafiq**

**Overview**

This project is focused on scraping data from IMDb for different movie categories, storing the information directly into an SQL database, and creating visualizations based on that data.

The project is structured into six main folders, each containing Python scripts that handle specific tasks. One folder is dedicated to Oscar awards data, and one is dedicated to most popular celebrities, while the others focus on different aspects of movie information and data processing.

**Technologies Used in Project:**

**Selenium:**

* I used Selenium for automating the browser to scrape data from websites like IMDb.
* It helped me navigate web pages, extract data using specific selectors, and interact with various web elements.

**WebDriver Manager:**

* To manage the browser driver, I used WebDriver Manager, which automatically downloads and configures the correct version of ChromeDriver.
* This eliminated the need for me to manually handle browser drivers.

**pyodbc:**

* I used pyodbc to connect my project to an SQL database.
* It allowed me to store and retrieve the scraped movie data in a structured format within the database.

**Matplotlib:**

* For visualizing the data, I used Matplotlib to create various types of graphs, such as bar charts and histograms.
* This helped me represent the movie data graphically for better understanding and analysis.

**Pandas (pd):**

* I relied on Pandas to manipulate and analyze the scraped data efficiently.
* With Pandas, I could clean, transform, and prepare the data both for visualization and storage in the SQL database.

**Folder Structure**

**1. Lowest Rated**

This folder contains scripts to scrape and analyze the lowest-rated movies on IMDb. The files in this folder include:

* **byAlpha.py:** Scrapes the lowest-rated movies and organizes them alphabetically.
* **byimdRating.py:** Scrapes data based on IMDb ratings.
* **byNo\_Rating.py:** Handles movies with no. of ratings.
* **byPopular.py:** Scrapes data for the lowest-rated movies but filters by popularity.
* **byR\_Time.py:** Scrapes the lowest-rated movies and organizes them by runtime.
* **byRel\_Date.py:** Scrapes and organizes movies by their release date.
* **lowestRated.py:** Main script that orchestrates scraping the lowest-rated movies.
* **lowVisualize.py:** Visualizes the scraped data using graphs or charts.
* **Sorting.py:** Contains functions for sorting the scraped movie data.
* **Index.txt:** This file contain the explanation of how the code running and purpose for my understanding. And if someone else use my code he/she can understand it.

**2. Top250Movies**

This folder contains scripts focused on scraping the top 250 movies from IMDb. And this folder also contain **nine** files as above explained.

**3. TopBoxOffice**

This folder focuses on scraping data related to the current box office movies. And contain one file **TopBox.py** that will tell us about a movie **Revenue/Gross** Rate Weekly and Total.

**4. PopularCeleb**

This folder contains scripts to scrape data about popular celebrities in the movie industry.

This folder also contain two files as above explained.

1. Alphabetic Order
2. Star Meter

**5. Top250TvShows**

This folder is dedicated to scraping the top 250 TV shows from IMDb. And this folder also contain **nine** files as above explained.

**6. Oscar-Award**

The **Oscar Awards** folder contains scripts designed to scrape and manage data related to the Academy Awards. This includes extracting information on various award categories (such as Best Picture, Best Director, Best Actor, etc.), listing the winners and nominees for each category, and recording the names of the movies, actors, and directors associated with these honors. The data is organized for easy access, allowing users to explore the history of Oscar nominations and victories in detail.

**Rest of the File that are not in any Six Folders**

1. **Explain.py:** This file contain also explanation of each file from each folder because all files have almost same code just the value and some column fields change But method is same so just explained all in 1.
2. **IMDB.docx:** This file contain the project requirement what is the project and how to make the project **what and how**.
3. **SQL\_1.sql:** This is the SQL file contain the tables name to see the particular file data in Table form.
4. **Project Documentation.docx:** This is the Project Documentation file you are currently reading.
5. **Dahboard.docx / .pdf:** Contain the graph detail what font style, size, color name, color code, chart style, used in this project.

**Data Scraping Methodology**

To scrape data from IMDb, I use the Selenium library. The key steps include:

**Setup:**

* I use WebDriver Manager to manage ChromeDriver installation automatically.
* Set up Chrome options, including a custom User-Agent to mimic a real browser.

**Navigating to the URL:**

* I use Selenium to open the IMDb website corresponding to the desired category (e.g., top 250 movies, lowest-rated and more).

**Data Extraction:**

* Here, I locate the relevant elements on the webpage (e.g., movie titles, ratings, release dates) using XPath or Class Name.
* And extract the required data fields for each movie.

**Data Storage:**

* In this step, I store the scraped data directly into a structured format (like tables) into SQL Database for easy manipulation and analysis.

**Data Visualization:**

* I use a libraries such as Matplotlib or Seaborn to visualize the data (e.g., bar charts for ratings, histograms for release years).

**Example of Scraping Code**

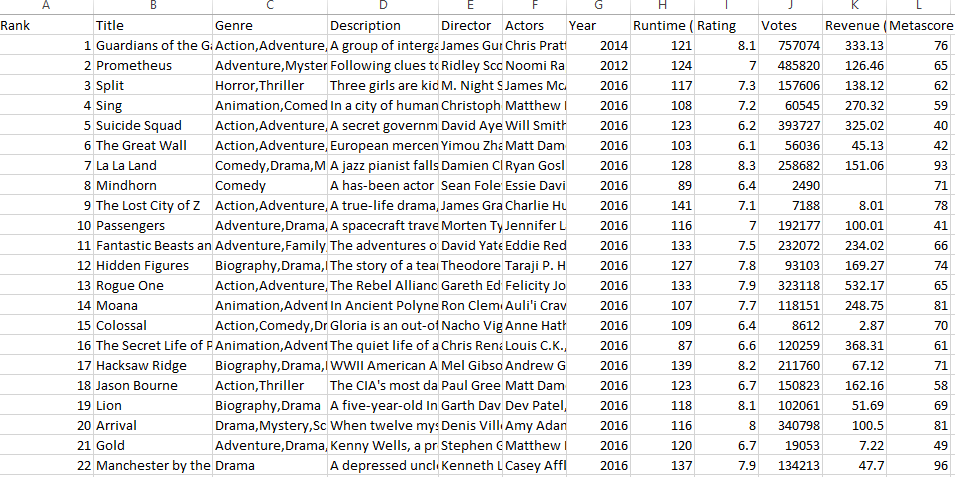
Below is a simplified example of how the scraping is implemented in one of the scripts

(e.g., lowestRated.py):



**SQL Database Data Sample**

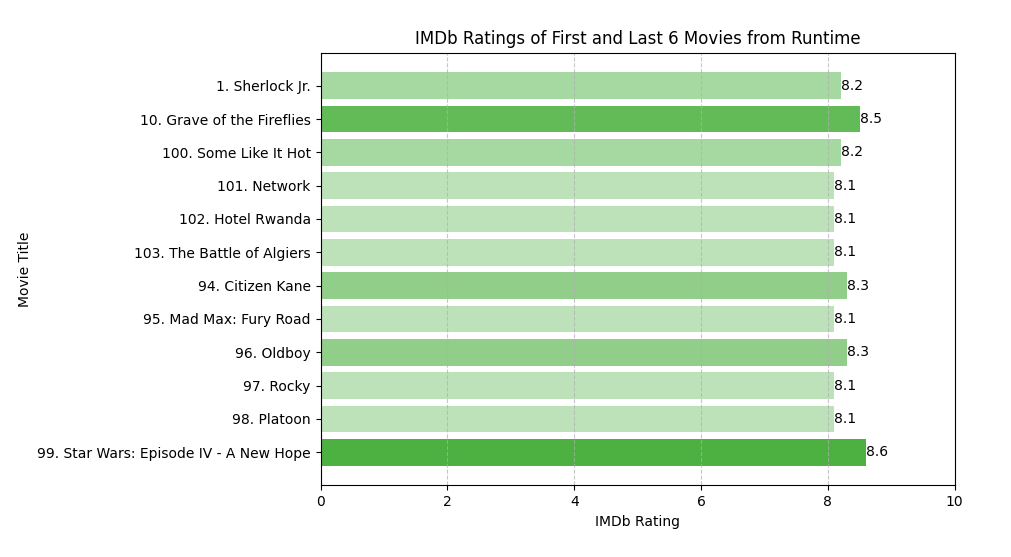
When the python script will execute, it will stored the data in an organized tables with defined columns and data types.

****

**Data Visualization Chart**

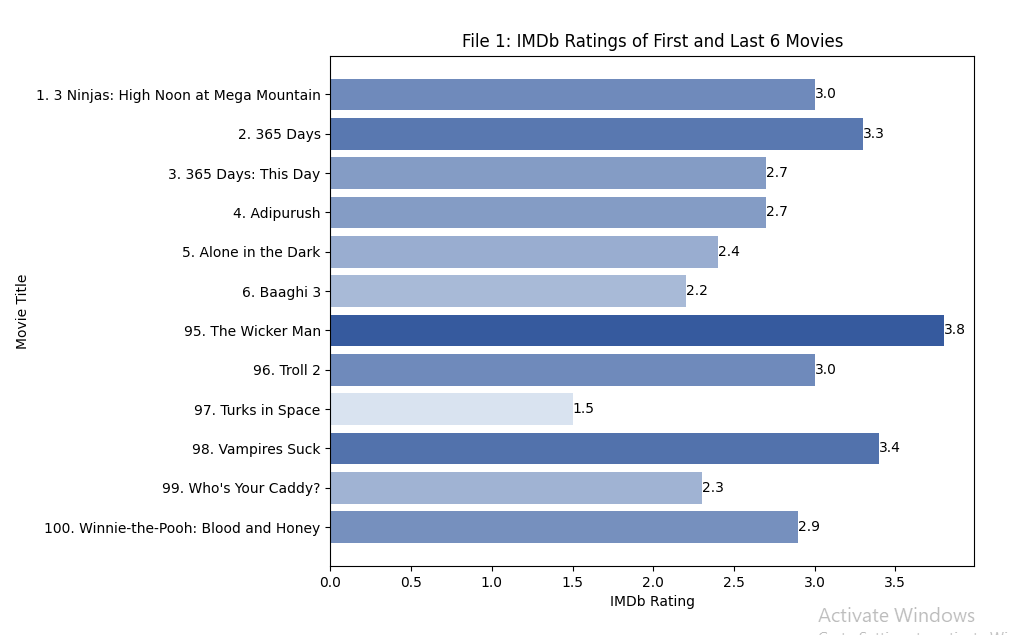
**Folder: 1 -** IMDB Top 250 Movies

This is the Chart of Top 250 Movies Sorted by Runtime and Y-axis contain the movies titles and X-axis contain the Rating values.

****

**Folder: 2 -** IMDB Top Low Rated Movies

This is the Chart of Top Lowest Rated Movies Sorted by alphabetic Order and Y-axis contain the movies titles and X-axis contain the Rating values.



**Conclusion**

In this project, I used Python to scrape and visualize data. Each folder has a specific purpose, and the scripts work together to give me useful information about movies, TV shows, Popular Celebrities, Top Box Office (US) and more on IMDb. I set up the project so that it smoothly collects, analyzes, and shows the data directly stored in SQL Database, and visualize Bar chart graph, helping me understand trends and patterns in IMDb rankings.