

| | |
|---|--|
| Project Title | IMDB 2024 Data Scrapping and Visualizations |
| Skills take away From This Project | Selenium, Python, Pandas, Streamlit, SQL, Data Cleaning, Data Analysis, Visualization, and Interactive Filter Applications. |
| Domain | Entertainment / Data Analytics |

Problem Statement:

This project focuses on extracting and analyzing movie data from IMDb for the year 2024. The task involves scraping data such as movie names, genres, ratings, voting counts, and durations from IMDb's 2024 movie list using Selenium. The data will then be organized genre-wise, saved as individual CSV files, and combined into a single dataset stored in an SQL database. Finally, the project will provide interactive visualizations and filtering functionality using Streamlit to answer key questions and allow users to customize their exploration of the dataset.

Business Use Cases:

1. **Top-Rated Movies:** Identify the top 10 movies with the highest ratings and voting counts.
 2. **Genre Analysis:** Explore the distribution of genres in the 2024 movie list.
 3. **Duration Insights:** Analyze the average duration of movies across genres.
 4. **Voting Patterns:** Discover genres with the highest average voting counts.
 5. **Popular Genres:** Identify the genres that dominate IMDb's 2024 list based on movie count.
 6. **Rating Distribution:** Analyze the distribution of ratings across all movies.
 7. **Genre vs. Ratings:** Compare the average ratings for each genre.
 8. **Duration Extremes:** Identify the shortest and longest movies in 2024.
 9. **Top-Voted Movies:** Find the top 10 movies with the highest voting counts.
 10. **Interactive Filtering:** Allow users to filter movies by ratings, duration, votes, and genre and view the results in a tabular DataFrame format.
-

Approach:

1. Data Scraping and Storage

- **Data Source:** IMDb 2024 Movies page ([link](#)).
- **Scraping Method:** Use Selenium to extract the following fields:
 - Movie Name
 - Genre
 - Ratings
 - Voting Counts
 - Duration
- **Genre-wise Storage:** Save extracted data as individual CSV files for each genre.
- **Combine Data:** Merge all genre-wise CSVs into a single DataFrame.
- **SQL Storage:** Store the merged dataset into an SQL database for querying and future analysis.

2. Data Analysis, Visualization, and Filtration

Interactive Visualizations

Using Python and Streamlit, create dynamic visualizations for:

1. **Top 10 Movies by Rating and Voting Counts:** Identify movies with the highest ratings and significant voting engagement.
2. **Genre Distribution:** Plot the count of movies for each genre in a bar chart.
3. **Average Duration by Genre:** Show the average movie duration per genre in a horizontal bar chart.
4. **Voting Trends by Genre:** Visualize average voting counts across different genres.
5. **Rating Distribution:** Display a histogram or boxplot of movie ratings.
6. **Genre-Based Rating Leaders:** Highlight the top-rated movie for each genre in a table.
7. **Most Popular Genres by Voting:** Identify genres with the highest total voting counts in a pie chart.
8. **Duration Extremes:** Use a table or card display to show the shortest and longest movies.
9. **Ratings by Genre:** Use a heatmap to compare average ratings across genres.
10. **Correlation Analysis:** Analyze the relationship between ratings and voting counts using a scatter plot.

Interactive Filtering Functionality

- Allow users to filter the dataset based on the following criteria:
 - **Duration (Hrs)**: Filter movies based on their runtime (e.g., < 2 hrs, 2–3 hrs, > 3 hrs).
 - **Ratings**: Filter movies based on IMDb ratings (e.g., > 8.0).
 - **Voting Counts**: Filter based on the number of votes received (e.g., > 10,000 votes).
 - **Genre**: Filter movies within specific genres (e.g., Action, Drama).
- Display the filtered results in a **dynamic DataFrame** within the Streamlit app.
- Combine filtering options so users can apply multiple filters simultaneously for customized insights.

Example Use Case:

- Users can filter for **Action Movies** with ratings above **8.0**, duration between **2-3 hours**, and voting counts greater than **50,000**, and the results will be displayed dynamically as a table.

Dataset

- Scraped IMDb data for 2024 movies, organized genre-wise.
- Columns:
 - Movie Name
 - Genre
 - Ratings
 - Voting Counts
 - Duration

Technical Tags

Languages: Python

Database: MySQL/PostgreSQL

Visualization Tools: Streamlit

Libraries: Pandas, Selenium, Matplotlib, SQLAlchemy, Seaborn



Project Deliverables

1. **SQL Database:** Contains the complete movie dataset.
2. **Python Scripts:** For data scraping, cleaning, merging, and database interaction.

3. **Streamlit Application:** Interactive dashboard showcasing visualizations, insights, and filtering functionality.
 4. **CSV Files:** Genre-wise datasets for each genre in the IMDb list.
 1. **Streamlit Application:** Interactive dashboard for real-time analysis.
-

Project Guidelines:

- **Follow coding standards:** Consistent naming conventions, modular code.
- **Data validation:** Ensure all data is accurate and complete.
- **Optimized queries:** Efficient SQL queries for large datasets.
- **Documentation:** Well-documented code and a detailed project report.

| | |
|---|---|
| Streamlit Doc | https://docs.streamlit.io/library/api-reference |
| Streamlit recording (Tamil) | Special Session for STREAMLIT Tamil |
| Project Live Evaluation |  Project Live Evaluation |
| GitHub Reference |  How to Use GitHub.pptx |
| Project Orientation (Tamil) | Project Orientation Session Recording |
| Web Scraping using Selenium- Special Session (Tamil) | Selenium- Special Session (Tamil) |
| Project Orientation 2 | Orientation Rec Tamil |

Project Evaluation metrics:

- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on **GitHub**. (Mandatory)
- You have to keep your **GitHub** repo public so that anyone can check your code. (Mandatory)
- Proper readme file you have to maintain for any project development (Mandatory)
- You should include basic workflow and execution of the entire project in the readme file on **GitHub**
- Follow the coding standards: <https://www.python.org/dev/peps/pep-0008/>
- You need to Create a Demo video of your working model and post in **LinkedIn** (Mandatory)

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.

Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday to Saturday (4:00PM to 5:00PM)

Booking link : <https://forms.gle/XC553oSbMJ2Gcfug9>

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

Note: This form will Open on Saturday and Sunday Only on Every Week

Timing: Monday-Saturday (5:30PM to 7:00PM)

Booking link : <https://forms.gle/1m2Gsro41fLtZurRA>

| Created By: | Verified By: | Approved By: |
|------------------|--------------|-----------------|
| Asvin Selvarajan | Shadiya | Nehlath Harmain |