IMDB MOVIE SUGGESTIONS APP

Project Overview:

The **IMDb Movie Suggestions Application** is a desktop tool that allows users to search for movies from IMDb's Top 250 list based on a selected genre. The app fetches movie data such as title, release year, runtime, rating, and number of votes. It also provides the ability to export the fetched movie data to an Excel file for further analysis. This project was developed using Python and integrates multiple libraries for web scraping, graphical user interface (GUI) development, and data export.

Features

- **Search Movies by Genre**: Users can select a genre from a dropdown list and fetch the top-rated movies of that genre from IMDb's Top 250 movies list.
- **Display Movie Data**: The movie data, including title, year, runtime, rating, and votes, is displayed in a clean and organized table format.
- **Export to Excel**: The displayed movie data can be exported to an Excel file, with a timestamped filename, for offline storage or further analysis.
- **User-Friendly GUI**: The application is designed with a simple and intuitive interface using `Tkinter`.

LIBRARIES AND SOFTWARE USED

The following libraries and software were used in the development of this project:

1. Python

The core programming language used to develop this application.

2. Tkinter

- **Purpose**: `Tkinter` is the built-in Python library for creating graphical user interfaces (GUIs).
- **Usage**: This library is used to design the main window, buttons, labels, dropdown menu, and table for displaying movie suggestions.

Key components used from `Tkinter`:

- **`Label`**: Used to display text and instructions.
- **`Button`**: Triggers actions like search and export.

- **`Entry`**: Allows the user to input text.
- **`ttk.Combobox`**: A dropdown for genre selection.
- **`Treeview`**: Displays movie suggestions in a table format.

3. Selenium

- **Purpose**: `Selenium` is a popular web scraping library that allows you to automate browsers and extract data from websites.
- **Usage**: This library is used to scrape IMDb's Top 250 movie list based on the selected genre.

Key components used from `Selenium`:

- **`webdriver`**: Initializes the browser (ChromeDriver in this case) and controls it for web scraping.
- **`By`**: Helps locate elements on the web page using various methods (e.g., XPath).
- **`Service`**: Manages the ChromeDriver service.
- **`Options`**: Configures browser options (e.g., enabling headless mode).
- **`WebDriverWait`** **and** **`expected_conditions`**: Used to wait for elements to load before interacting with them.

4. pandas

- **Purpose**: `pandas` is a powerful data manipulation library, widely used for handling structured data in Python.
- **Usage**: It is used to store the fetched movie data in a structured format (DataFrame) and export it to Excel.

Key components used from `pandas`:

- **`DataFrame`**: A tabular data structure used to hold and manipulate movie data.
- **`to_excel()`**: Exports the DataFrame to an Excel file for saving the movie data.

5. openpyxl

- **Purpose**: `openpyxl` is a library used by `pandas` to read and write Excel files.
- **Usage**: Required for the functionality of exporting movie data to Excel.

6. datetime

- **Purpose**: The `datetime` module provides functions to manipulate dates and times.
- **Usage**: Used to generate a unique filename for the Excel export by appending a timestamp to the filename.

7. ChromeDriver

- **Purpose**: `ChromeDriver` is the WebDriver used by Selenium to interact with the Chrome browser.
- **Usage**: It is used in conjunction with Selenium to automate browsing and scraping IMDb data.

Application Workflow

1. **Launch the Application**: Start the app, which will open a graphical window built using `Tkinter`.
2. **Select Genre**: Choose a genre from the dropdown list, which includes genres such as Drama, Comedy, Action, Horror, etc.
3. **Click on "Search"**: Once a genre is selected, clicking the "Search" button will trigger the scraping process.
4. **Display Results**: The application will scrape IMDb's Top 250 list, filter by the selected genre, and display the movie titles along with their release year, runtime, rating, and number of votes.
5. **Export to Excel**: Users can click on the "Export" button to save the movie data to an Excel file with a timestamped filename.

Example Usage

1. **Select Genre**: Choose a genre, such as **"Comedy"**, from the dropdown.
2. **Search Movies**: Click the **Search** button, and the app will fetch movie suggestions of that genre from IMDb's Top 250.
3. **View Movie Data**: The movie titles, along with the year, runtime, rating, and votes, will be displayed in a table format.
4. **Export Data**: The **Export** button allows the user to save the movie data in an Excel file, which will be saved in the current directory with a unique name based on the current timestamp.

Installation

To use the IMDb Movie Suggestions Application, follow these steps:

1. **Install Python**: Make sure Python is installed on your machine.
2. **Install Required Libraries**: Install the necessary Python libraries by running the following command:
3. `pip install selenium pandas openpyxl`
4. **Download ChromeDriver**: Download the appropriate version of `ChromeDriver` from here based on your Chrome version. Ensure that `chromedriver` is accessible from your system's PATH or specify its location in the script.
5. **Run the Script**: After installing all dependencies, run the Python script. The application window will open, allowing you to interact with it.

Code Structure

The project structure is simple and can be understood in the following way:

```
IMDb_Movie_Suggestions/
│
├── IMDb Movie Suggestions App.py  # Main application script
├── output_{timestamp}.xlsx  # Exported movie data (generated during runtime)
├── README.md                # Project documentation (this file)
```

Troubleshooting

- **Selenium Errors**: If you encounter issues while scraping IMDb, ensure that your version of `ChromeDriver` is compatible with your installed version of Chrome.
- **No Data Displayed**: If no movie data is displayed after clicking "Search," ensure that your internet connection is stable and that IMDb's layout hasn't changed (which could break the scraping logic).
- **Excel Export Issues**: Ensure that `pandas` and `openpyxl` are properly installed for the export functionality to work.

Conclusion

This IMDb Movie Suggestions Application provides a simple yet powerful tool for discovering top-rated movies by genre, displaying the results in an intuitive GUI, and allowing users to export the data for further analysis. The integration of `Selenium`, `Tkinter`, and `pandas` ensures that the app is both functional and user-friendly.