Movie Sentiment Explorer

Advanced Computer Programming

Group 8

CONTENTS

Introduction

Movie Sentiment Explorer is a web application designed to help users decide what movies to watch by analyzing real IMDb user reviews. It uses AI-based sentiment analysis to determine whether reviews are generally positive, negative, or neutral. Movie details such as posters and descriptions are fetched using the OMDb and TMDb APIs, while actual reviews are collected through IMDb scraping. This tool provides a quick and clear summary of public opinion, making it easier for users to avoid poorly rated films and discover highly recommended ones.

Main Features

- Search Movie Info

Users can type a movie title to instantly fetch details like poster, genre, plot, and release year.

Example: Searching “Inception” shows its poster, plot summary, director, and more — all using the OMDb and TMDb APIs.

- Sentiment Analysis on IMDb Reviews

Instead of just showing star ratings, the app reads actual review texts and uses AI to detect if people really liked the movie — or not.

Example: If most reviews say “boring” or “not worth it,” the sentiment score reflects that — even if the rating looks okay. Why it matters: You’re not seeing just numbers — you’re seeing real emotions.

System Architecture

- Sentiment Chart (Chart.js)

All those review sentiments are visualized in a clean pie chart. You instantly see how many users were happy, neutral, or disappointed.

Example: A movie might show 70% positive, 20% neutral, and 10% negative — much more useful than just “7/10”.

Why it matters: Visual feedback helps users make faster, smarter decisions.

- Export Options (PDF, CSV)

After exploring a movie, users can export the review data and analysis in PDF or CSV format.

Example: Download a summary for your movie night group, class project, or just to save it for later.

Why it matters: It’s not just about browsing — it's about taking insights with you

- UI Theme Toggle

Choose between light mode and dark mode for comfort or aesthetic.

Example: Watching at night? Dark mode makes it easier on the eyes.

Why it matters: It feels like a modern app — personalized to you.

System Architecture

1. Frontend (User Interface)

- Built with HTML, Tailwind CSS, and JavaScript
- User enters a movie name and interacts with search, export, and theme toggle features

2. Backend (Flask)

- Handles form submissions, routes, and logic
- Communicates with APIs and manages sentiment analysis

3. External APIs

- OMDb + TMDb APIs: Fetch movie details, poster, genre, etc.
- IMDb Scraper: Collects real user reviews for the searched movie

System Architecture

Feature	OMDb	TMDb	IMDb Scraping
Movie title, plot, genre	✅	✅	❌
IMDb rating	✅	❌	❌
Director, release year	✅	✅	❌
Poster (main source)	✅	✅	❌
High-res backup poster	❌	✅	❌
Review text	❌	❌	✅
Sentiment analysis input	❌	❌	✅

Each API plays a unique role in making the project complete. OMDb handles the core movie info, TMDb enhances visuals, and IMDb scraping powers the sentiment analysis. By combining all three, the site doesn't just show facts — it shows how people feel about each movie.

System Architecture

4.Sentiment Analysis (VADER)

- Processes scraped review texts
- Returns compound scores from -1 (negative) to +1 (positive)

5. Output (User View)

- Movie info displayed on the page
- Sentiment score shown as a number
- Pie chart (using Chart.js) shows sentiment breakdown visually
- Users can export data as PDF or CSV

Key Technologies

🧩 Flask (Python Web Framework)
Used to build the backend of the website — handles routing, user input, and connects everything together.
🌐 BeautifulSoup (Web Scraping)
Used to scrape real user reviews directly from IMDb, since that data isn't available through APIs.
🧠 NLTK + VADER (Sentiment Analysis)
VADER is a pre-trained sentiment analyzer in NLTK that reads the review text and gives a compound score showing how positive or negative it is.
📊 Chart.js (Data Visualization)
Used to create an interactive pie chart that visually shows the percentage of positive, neutral, and negative reviews for each movie.
📦 Dataclasses (Python Feature)
Helps structure and organize movie data neatly in the backend code.
🔍 Regex (Pattern Matching)
Used to clean and extract useful parts from raw scraped HTML or text content, like filtering out unwanted characters or links.

Live DEMO!

Conclusion

In this project, We combined multiple technologies to create a smart and user-friendly movie sentiment analysis website. Flask handles the backend logic, BeautifulSoup scrapes real reviews from IMDb, and VADER analyzes the emotional tone of those reviews. The results are visualized using Chart.js to give users a clear picture of how people feel about each movie. By integrating OMDb and TMDb APIs, the site also provides complete movie details. Altogether, this system turns raw data into useful insights, helping users make better movie-watching decisions.

Thank You

https://github.com/ACP-Final-Group12/ACP_Final.git