WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Movie Mania

Selvaraj A[1], Ruban Kumar V[2], Nirendrasethupathi A[3], Praveen G [4]

[1]Guide, Department of Information Technology, K.L.N. College of Engineering, Sivaganga – 630 612, Tamil Nadu, India,
[2,3,4]Student, Department of Information Technology, K.L.N. College of Engineering, Sivaganga – 630 612, Tamil Nadu, India
Selvax25@gmail.com[1], ruban2572004@gmail.com[2], nirendrasethupathi@gmail.com[3], kieronpraveen@gmail.com[4]

ABSTRACT

MovieMania is a comprehensive movie discovery and analysis platform designed to transform how audiences search, evaluate and explore films. The platform integrates a vast, up-to-date movie database with advanced filtering options that allow users to search for films by genre, release date, cast and more. In addition, MovieMania aggregates ratings from leading sources such as IMDb, Rotten Tomatoes and Metacritic to present reliable and unbiased recommendations. The system also features a dual-review model, combining detailed user reviews with expert critic insights to provide diverse perspectives on movie quality and impact.The platform is supported by a robust technical architecture that ensures smooth performance and scalability. MovieMania combines functionality with innovation to provide users an engaging and informative movie discovery experience. To fuance user engagement, the system incorporates personalized recommendations based on viewing history and preferences.

Keywords: Movie Recommendation **,** Personalized Movie Suggestions, User and Critic Reviews,Multisource Ratings Integration, Enhanced Movie Discovery.

1. INTRODUCTION

MovieMania is a next-generation movie discovery and analysis platform aimed at redefining the way users explore, evaluate, and engage with films. The platform brings together a comprehensive movie database, intelligent filtering tools, multi-source rating aggregation, and a dual-review system that combines public opinion with expert critiques. With a robust backend powered by Java and PostgreSQL and a responsive frontend built using HTML and CSS, MovieMania ensures a seamless user experience. The system leverages AWS for scalable cloud deployment and integrates advanced analytics and personalized recommendation features, making it both powerful and user-centric

2. LITERATURE REVIEW

[1] Zhuang, Li, Feng Jing, and Xiao-Yan Zhu. "Movie review mining andsummarization." In Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 43-50. 2006.

[2] Berman, Jeffrey. Mad Muse: The Mental Illness Memoir in a Writer's Life and Work. Emerald Publishing Limited, 2019.

[3] Putri, Risma Amelia, Syafa Zhafira Putriyanda, Nur Intan Permatasari, and Fakhirah Nur Inayyah. "DYNAMICS OF PSYCHOLOGY IN THE MOVIE â€ œ I THOUGHT YOU WAS HOMEâ€." In Prosiding Seminar Nasional Inovasi Pendidikan. 2022.

[4] Mao, Regina, Fransiskus M. Separ, and Yuliana MDK Kara. "MANIC DEPRESSIVE DISORDER OF THE MAIN CHARACTER AS SEEN IN Mr. JONES MOVIE (PSYCHOLOGYCAL APPROACH)." Lantern: Journal of Language and Literature 10, no. 2 (2024): 44-50.

[5] Anderson, Sal, and Dolly Sen. "'Film Is Psychosis': Filmmakers with Lived Experience." (2023).

[6] O'Pray, Michael. "Movies, mania and masculinity." Screen 23, no. 5 (1982): 63-71.

3. METHODOLOGY

The Movie Mania application was developed using HTML, CSS, and JavaScript. It employs RESTful communication with the TMDB API to fetch movie details in JSON format. The system follows a modular design approach separating concerns into distinct responsibilities such as data fetching, user input handling, and DOM manipulation. Tools like DFD and UML were utilized for modeling and analysis. Evaluation was performed based on software metrics such as scalability, maintainability, and usability.

4. MODELING AND ANALYSIS

1. **Introduction**

This document provides a structured modeling and analysis of the **Movie Mania** web application, a client-side app that fetches and displays movie data from the TMDB API based on popularity and user search queries.

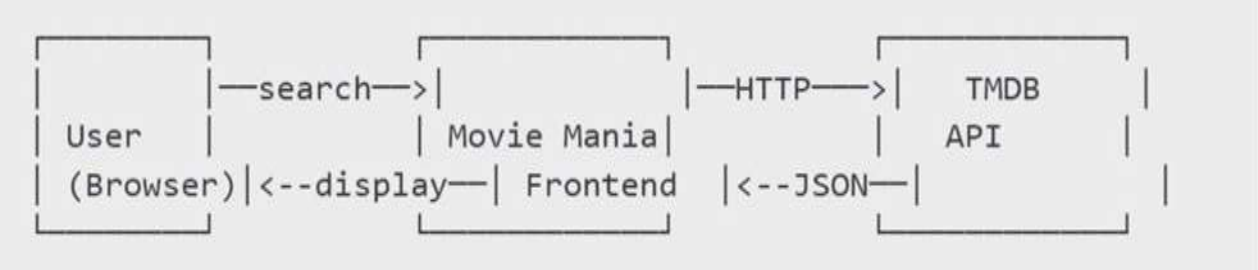
2. **Functional Overview & Requirements Primary features:**

- Display top popular movies on load
- Allow user to search for movies by title
- Show movie details (poster, title, rating, overview)

Actors:

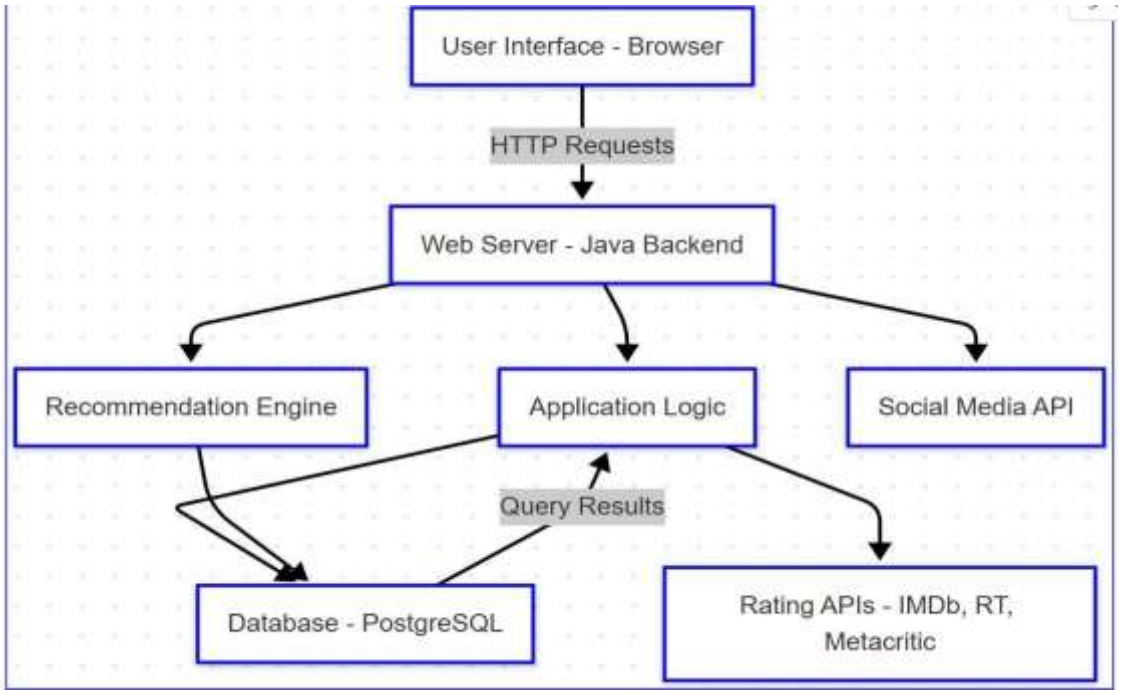
- **User:** interacts via browser UI
- **External TMDB API:** provides movie data

3. ***System Context Diagram***



Description: The user submits a search or views popular movies. The frontend fetches data via HTTP from TMDB and renders it in the UI.

4. **Architecture diagram*]***



5. Result and Discussion

Results

The "Movie Mania" application successfully fulfills its goal of allowing users to discover and search for movies. The primary features implemented include:

- ***Homepage Display**: When the application loads, it fetches and displays a list of popular movies using the TMDb API (/discover/movie?sort_by=popularity.desc).*
- ***Search Functionality**: Users can search for specific movies using the search input, which queries the API endpoint (/search/movie) dynamically.*
- ***Visual Movie Cards**:*
  - *Each movie is presented as a card containing its **poster**, **title**, **average rating**, and **overview**.*
  - *Ratings are color-coded:*
    - *Green for high-rated (>= 7.5),*
    - *Orange for average-rated (7 to 7.4),*
    - *Red for low-rated (< 7).*
- ***Responsive UI**: The layout adapts to smaller screens, ensuring usability on both desktops and mobile devices.*
- ***Smooth UX Features**: Hover effects reveal movie overviews, and the design uses modern aesthetics with Poppins font, subtle shadows, and color themes. **Discussion***

This project demonstrates practical integration with a third-party REST API, showing competence in:

- *Fetching and parsing JSON data asynchronously using fetch and async/await.*
- *DOM manipulation and dynamic content rendering based on API responses.*
- *Applying conditional logic to enhance the user interface (e.g., color-coded ratings).*
- *Designing a responsive UI using CSS Flexbox and media queries.*

However, there are some areas for improvement:

- ***API Key Security**: The API key is hardcoded in the JavaScript file, which poses a security risk. For production, this should be moved to a backend server.*
- ***Error Handling**: Currently, the app doesn't account for API errors or empty search results. Implementing try-catch blocks and user feedback (e.g., "No results found") would improve robustness.*
- ***Pagination**: Only the first page of results is fetched. Adding pagination or infinite scroll would improve usability for larger datasets.*

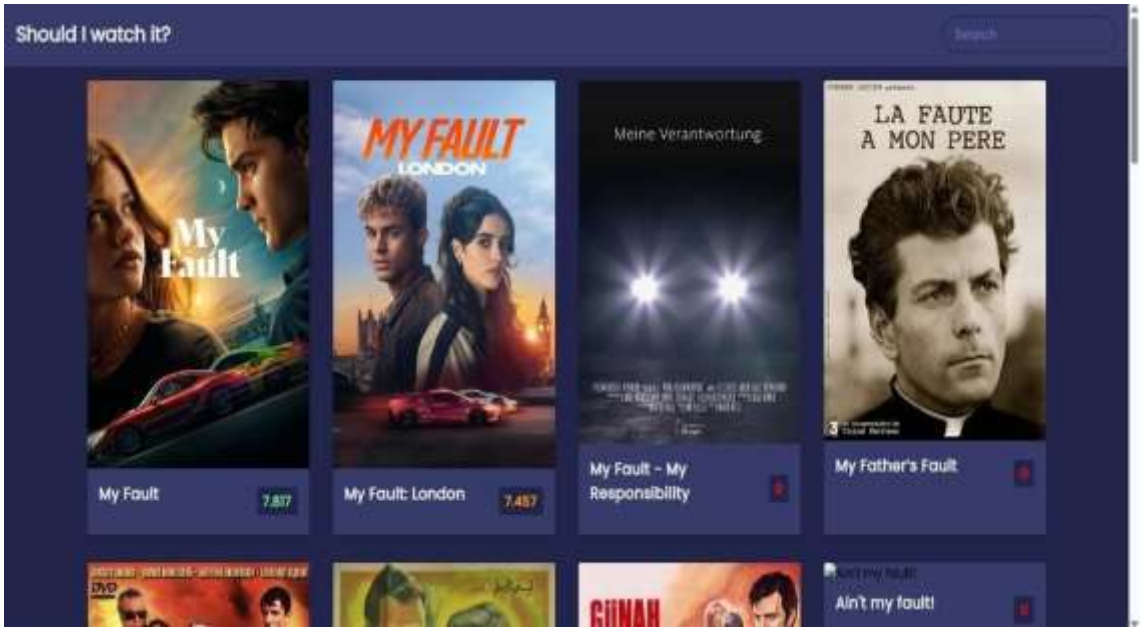


Fig 1 Landing page

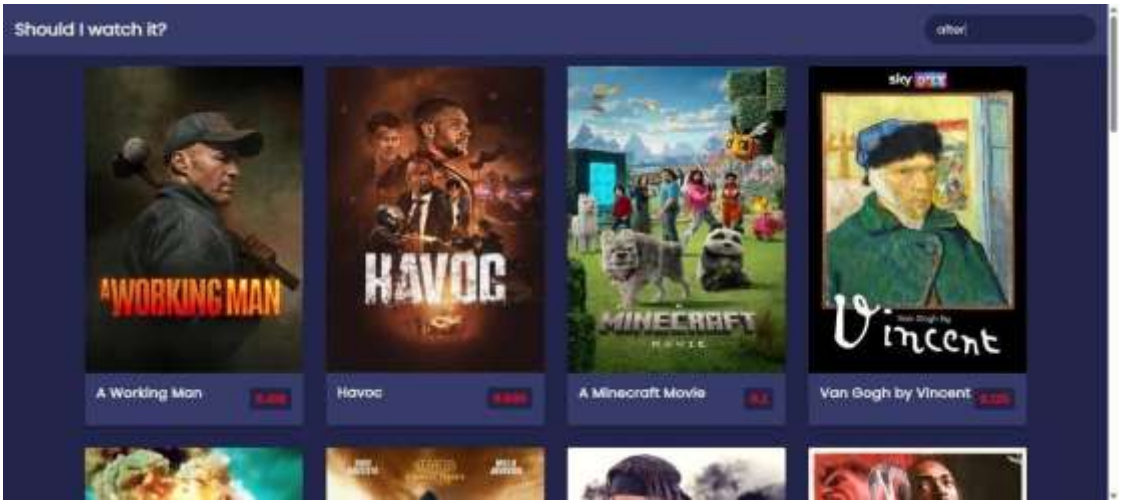


Fig 1.2 Movies Search Option

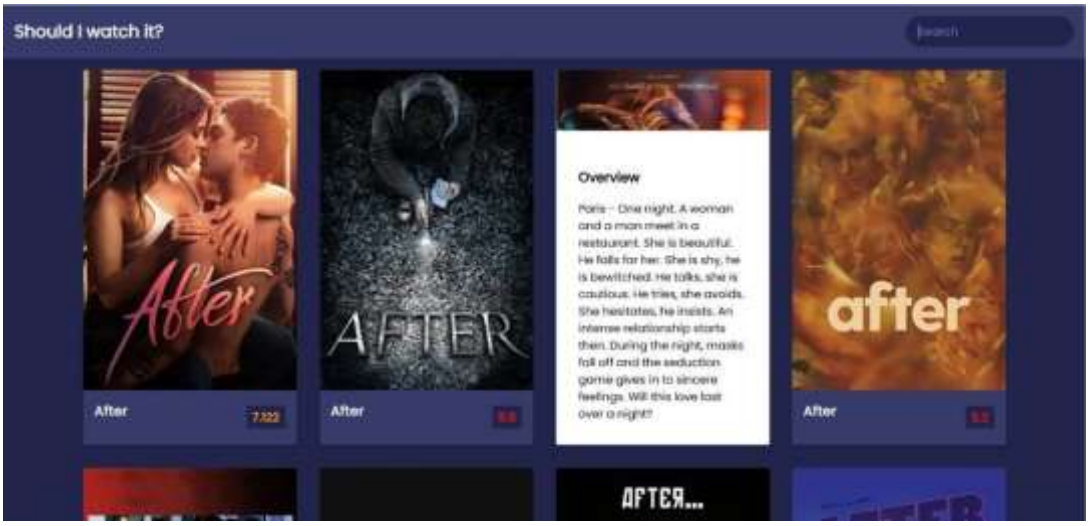


Fig 1.3 Description About Movie

6. CONCLUSION

The "Should I Watch It?" web application effectively addresses the challenge of content overload faced by modern streaming audiences. By integrating real-time data from the TMDb API with a responsive and user-friendly front-end built using HTML, CSS, and JavaScript, the application provides a seamless way for users to discover movies based on popularity and personal search preferences. The use of a color-coded rating system enhances decision-making, while dynamic content rendering ensures a smooth and interactive experience across devices. This project not only demonstrates the power of combining external APIs with core web technologies but also showcases how thoughtful UI/UX design can significantly improve user engagement and satisfaction. .

7. FUTURE ENHANCEMENT

While the current version of "Should I Watch It?" offers a solid foundation for movie discovery, there are several opportunities for future

development to enhance functionality and user experience:

1. User Authentication and Personalization Integrate login functionality to allow users to create accounts, save favorite movies, and receive personalized recommendations based on watch history and preferences.

2. Genre and Filter Options Add advanced filtering capabilities such as sorting by genre, release year, language, or user ratings to refine search results more effectively.

3. Recommendation Engine Implement a basic collaborative or content -based recommendation algorithm to suggest movies tailored to

individual user interests.

4. Watchlist and Favourites Allow users to create and manage a watchlist or favorites list that persists across sessions

Acknowledgements

We extend our sincere gratitude to Dr. P. Ganesh Kumar and Ms. A. Selvaraj for their invaluable guidance throughout this research. We also thanks K.L.N college of Engineering for providing the necessary resoures and support for this project.

REFERENCES

[1]Chaovalit, Pimwadee, and Lina Zhou. "Movie review mining: A comparison between supervised and unsupervised classification approaches." In Proceedings of the 38th annual Hawaii international conference on system sciences, pp.112c-112c. IEEE, 2005.

[2].Zhuang, Li, Feng Jing, and Xiao-Yan Zhu. "Movie review mining and summarization." In Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 43-50. 2006.

[3] Berman, Jeffrey. Mad Muse: The Mental Illness Memoir in a Writer's Life and Work. Emerald Publishing Limited, 2019.

[4[Putri, Risma Amelia, Syafa Zhafira Putriyanda, Nur Intan Permatasari, and Fakhirah Nur Inayyah. "DYNAMICS OF PSYCHOLOGY IN THE MOVIE â€ œ I THOUGHT YOU WAS HOMEâ€." In Prosiding Seminar Nasional Inovasi Pendidikan. 2022.

[5] Mao, Regina, Fransiskus M. Separ, and Yuliana MDK Kara. "MANIC DEPRESSIVE DISORDER OF THE MAIN CHARACTER AS SEEN IN Mr. JONES MOVIE (PSYCHOLOGYCAL APPROACH)." Lantern: Journal of Language and Literature 10, no. 2 (2024): 44-50.

[6] Anderson, Sal, and Dolly Sen. "'Film Is Psychosis': Filmmakers with Lived Experience." (2023). [7]O'Pray, Michael. "Movies, mania and masculinity." Screen 23, no. 5 (1982): 63-71.