MORDERN APPLICATION DEVELOPMENT

(JAVA SPRINGBOOT)

FilmCrictic

(Movie Review Platform)

TEAM NO: 415

BHADHRINATH U – 20BCE0183

JAIJEEVAALSHREE N – 20BCE0010

CHIBHIRAJ K – 20BCT0029

DEEKSHITH REDDY B U – 20MIC0058

Table of Contents

1. INTRODUCTION

1.1 OVERVIEW

The movie review system project makes use of several technologies to create a comprehensive application, including Spring Boot, React, MongoDB, Docker, and Kubernetes. An outline of the project is given below: Strong RESTful APIs are provided by the Spring Boot-built backend for handling requests and controlling the business logic of the application. React enables dynamic and interactive UI components on the frontend, improving the user experience. Movie data, user data, and review data are stored and retrieved quickly using the NoSQL database MongoDB. While Kubernetes orchestration manages scalability, fault tolerance, and load balancing, making the application highly available and scalable, Docker containerization provides consistency and makes dependency management simple. These technologies combine to produce a thorough movie review system with seamless backend, frontend, and database integration, providing a positive user experience and effective application.

1.2 PURPOSE

The project's goal is to give users a platform where they can browse, evaluate, and share their thoughts on films. The project seeks to provide a user-friendly application that enables users to read and contribute reviews, find new films, and interact with a community of movie aficionados. The programme develops a community of movie buffs who can communicate, exchange ideas, and have discussions about their preferred films, resulting in a lively and engaged user base. Users can choose films that suit their interests and preferences by using the system's personalised movie suggestions, which are based on user choices and viewing history. In general, the project's goals are to improve users' movie-watching experiences and to create a lively community.

2. LITERATURE SURVEY

S.no.	Title	Authors (Year of Release)	Proposed Work
1	Web Development with ReactJS and Spring Boot	Harikrishna V Holla. (2022)	ReactJS uses components. ReactJS components are structures. Rendering react library creates DOM trees for each component. Render generates a tree from the bottom up recursively and creates intermediate DOM. After assembling intermediate DOMS creates HTML DOM. JSX-based ReactJS uses XML. JSX creates DOM trees. Components are XML nodes. JSX manages property-occasion handler relationships. React works with JavaScript and HTML without JSX. JSX simplifies web development.

			As indicated above, each component is a node and can have child nodes with simpler and broken down components. JSX untangles intermediate nodes. Each component has three states—Unmounted, Update, and Mounted. Rendering creates trees in the mounting state. Only when unmounted or mounted, the component goes to the Update state. The DOM is updated when the view changes. Unmounting occurs after state update.Dependencies drive Spring Boot. Imported dependencies load JPA repository JAR files. Spring Boot uses @SpringBootApplication. Project main class uses it. This Spring framework annotation combines @Configuration, @EnableAutoCOnfiguration, and @ComponentScan. The project searches for @Configuration classes and creates beans for them. IOC containers store these beans in JVMs. Based on bean scope. The dispatcher servlet and default handler mapping are configured automatically by these beans.
2	Load Balancing using Docker and Kubernetes: A Comparative Study	Prajval Mohan, Tejas Jambhale, Lakshya Sharma, Simran Koul, Simriti Koul. (2020)	The main goal of the Value-Based Allocation of Docker Containers was to find out what the main goal of docker containers is. Since the number of public cloud providers is growing quickly, they have added containers as a service (CaaS) to their offerings. This is why Docker is so famous. Docker is a piece of software that lets Linux containers run in a separate environment on a host. A Portable Load Balancer for Kubernetes Cluster says that Linux containers have become more popular because they are small and easy to move. These days, a lot of online services are set up as groups of

			containers. The writers of this paper have focused on Kubernetes Clusters. But Kubernetes needs cloud companies to handle load balancing. The authors suggested a portable load balancer that could be used in any setting. This would make it easier to move web services. This was done with the help of the Internet Protocol Virtual Server (IPVS) in the Linux kernel. The product made a portable web service better without hurting its efficiency. Distributed computing gives users near-instant access to what seems like an infinite number of resources and gives expert companies the chance to offer complex data technology systems as a service to their clients. By sharing assets through virtualization of the basic physical basis, suppliers can make more money through economies of scale and multiplexing. But because cloud platforms are so big and change so quickly, cloud expert co-ops have to deal with a lot of new problems.
3	A Movie Recommender System: MOVREC	Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta. (2015)	There are various approaches proposed in various research papers listed below. These approaches are often combined in Hybrid Recommender Systems. An earlier study by Eyjolfsdottir et. al for the recommendation of movies through MOVIEGEN had certain drawbacks such as , it asks a series of questions to users which was time taking . On the other hand it was not user friendly for the fact that it proved to be stressful to a certain extent. Keeping in mind these shortcomings, we have developed MovieREC, a movie recommendation system that recommends movies to users based on the information provided by the users themselves. In the present study, a user is given the option to select his choices from a set of

			attributes which include actor, director, genre, year and rating etc. We predict the users' choices based on the choices of the previous visited history of users. The system has been developed in PHP and currently uses a simple console-based interface.
4	Comparison of Java web application frameworks	Diman Salih (2019)	The reason of this project has been to assist web developers to see some well-known web application of Java frameworks closely in literature review: Spring, Struts, JSF and Tapestry through the contrasts. The purpose has also been to identify most appropriate web application group of these frameworks. In doing so, many steps have been followed in literature review. They have done design and implementation for two of the frameworks: Struts and JSF then Testing for both systems. In the first step he explained in some detail about some of the most used underlying technologies in Java web application frameworks. Then explored how they vary in conforming to the design patterns and applying their lifecycle to know in depth about their infrastructure mechanisms. Then he selected five web features for comparison between the four frameworks: Validation, Navigation rules, Internationalization, IoC and Ajax.
5	Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example	Kavya Guntupally, Ranjeet Devarakonda, Kenneth Kehoe. (2018)	The application is designed using three main components (as shown in figure), a frontend UI form (to capture user entered data), API (which accepts HTTP/S requests either from a UI form or command call, like curl or Postman) and the database (where the data is stored). The UI is a simple form with a set of fields that describes the data quality issue for the affected data. Once the form is submitted, the data is validated

			(check for required fields and expected formats). If valid, then it is sent as JavaScript Object Notation (JSON) request to the REST API using a resource URL. The API validates the incoming request against the entity fields required to save, update, or delete the data in the database. If the data is valid and the operation is successful, the API returns a HTTP response back to UI with success message and standard status code.

2.1 Existing problems

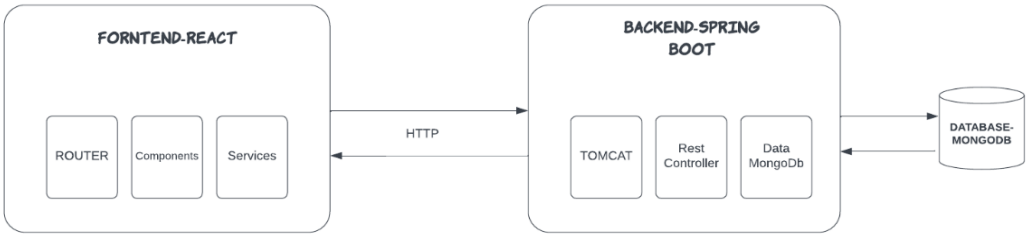
Limited Movie Selection: Users can discover that the movie database does not contain all the films they are looking for, particularly if the database is not updated frequently. This may reduce the variety of options offered to users. Reviews that are contradictory or biased may be encountered by users, which may influence how they decide. Reviews that lack credibility or are biased in some way tend to contain false or misleading information. Lack of User Engagement: The platform may have trouble attracting and keeping active users without a robust user engagement strategy. Limited user engagement and participation can impede the community's expansion and vitality.

2.2 Proposed Solution

Implementing a thorough movie data integration and validation system is one suggested remedy for the issues now present in the movie review system. To ensure a large and current selection of films in the database, our solution integrates several trustworthy movie data sources. Release dates, genres, and ratings for films would all be continually updated by the system. It is possible to design a two-step validation method to address the problem of inconsistent or biased reviews. First, suspect trends, spam, or biased reviews can be found using an automated review analysis programme. Second, a community-driven moderation system can be set up, allowing active users with high credibility scores to examine and confirm the reviews that have been reported. This strategy ensures a review system that is more dependable and trustworthy. Gamification components, such as leaderboards, awards, and badges, can be added to increase user engagement. Users can accumulate points by posting reviews, participating in conversations, and connecting with the community, which encourages competitiveness and rewards. Additionally, personalised movie suggestions can be provided by using algorithms that leverage user preferences, viewing history, and verified reviews. Combining these approaches enables the movie review system to offer consumers a full and enjoyable movie review experience by offering a diverse movie selection, reliable reviews, enhanced user engagement, and personalised suggestions**.**

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/software designing

Hardware:

- PC/ Laptop

Software:

- React
- Axios
- Visual studio Code
- Maven
- Jdk – Java 17
- IntelliJ IDE
- MongoDb Atlas

4. EXPERIMENTAL ANALYSIS

User Behaviour Analysis

- **User Interaction Patterns**: Analysing how users navigate the website, access movie listings, watch trailers, and submit reviews.
- **Load Testing**: Simulating heavy user loads to evaluate the system's performance under stress and ensure it can handle concurrent user requests effectively.

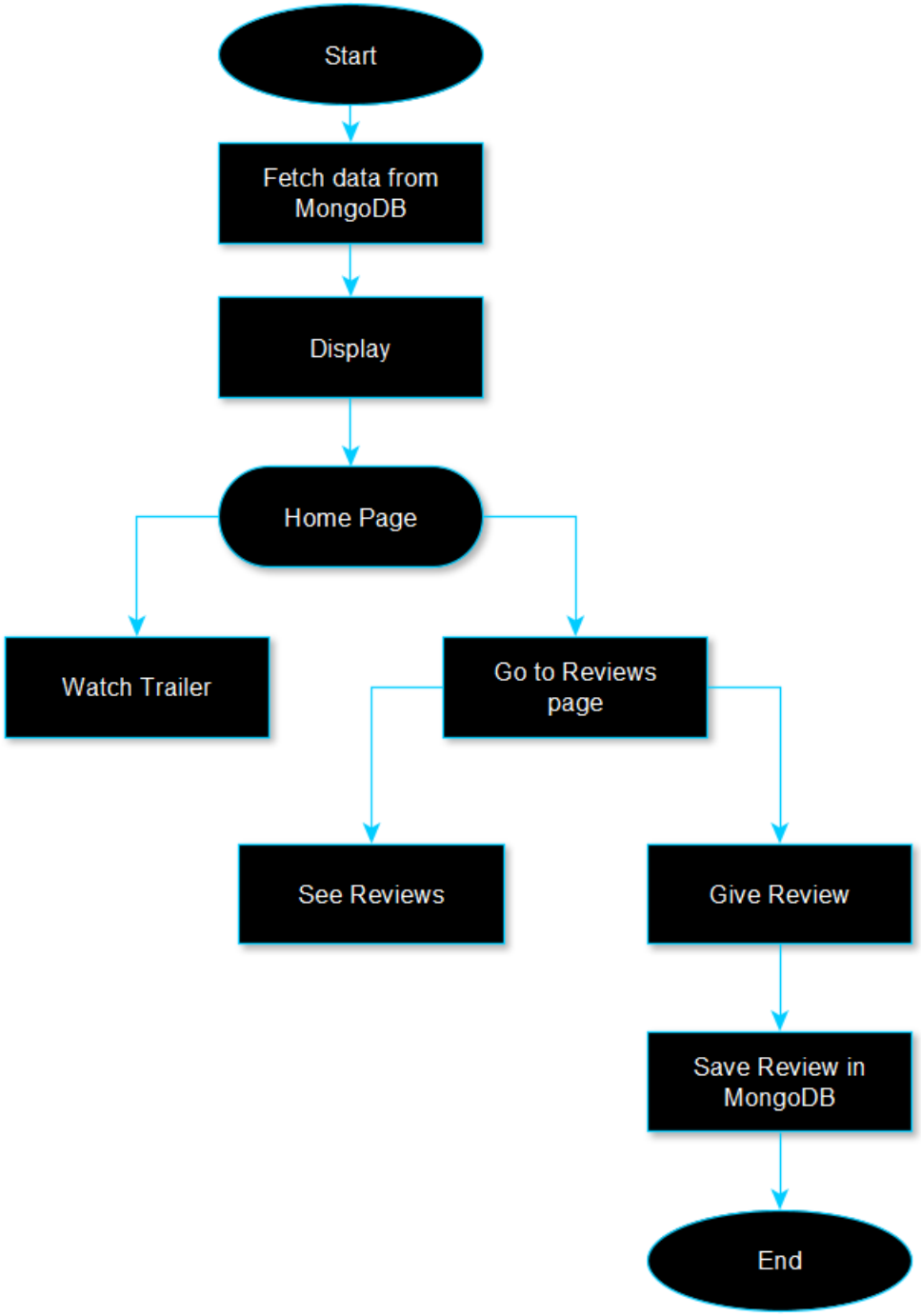
Performance Analysis

- **Response Time**: Measuring the system's response time for different operations, such as loading movie listings, retrieving reviews, and submitting reviews.
- **Error Tracking**: Monitor and analyse system errors or exceptions to identify any critical issues that may affect user experience.

Error Analysis and Bug Tracking

- **Error Logs and Reports**: Tracking and analysing error logs to identify common errors, exceptions, or bugs and addressing them promptly to improve system stability and user experience.
- **Debugging and Troubleshooting**: Investigating reported issues, conducting root cause analysis, and implementing necessary fixes to enhance system reliability.

5. FLOWCHART



6. RESULT

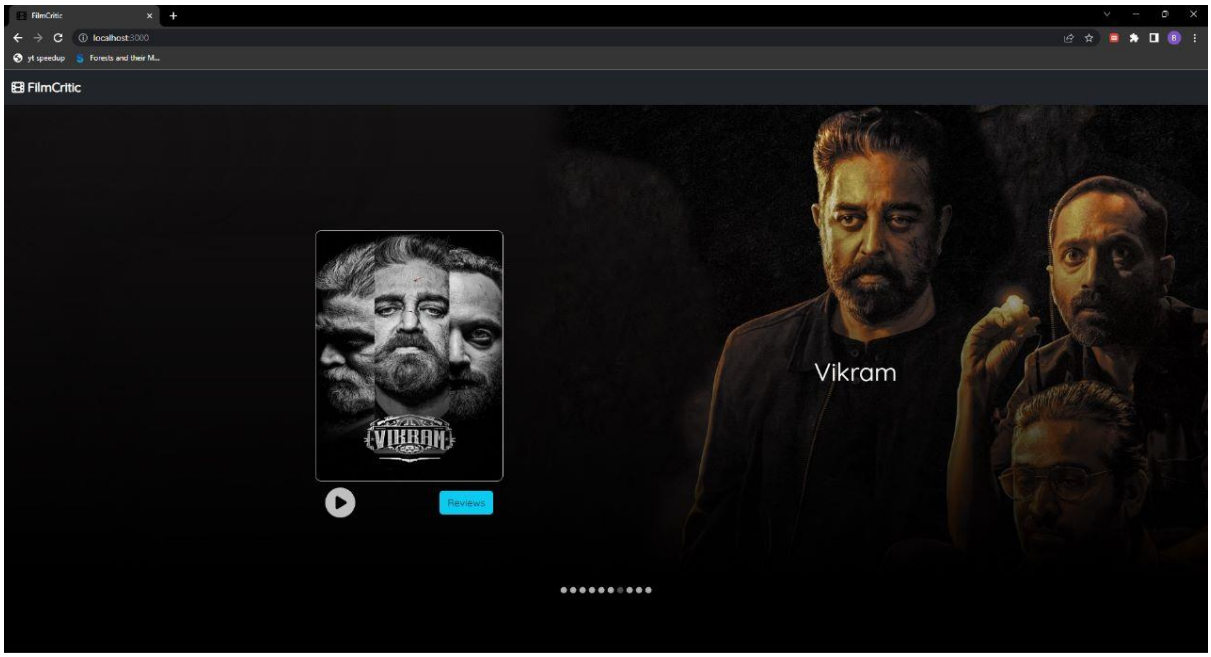


Fig 1: Home page

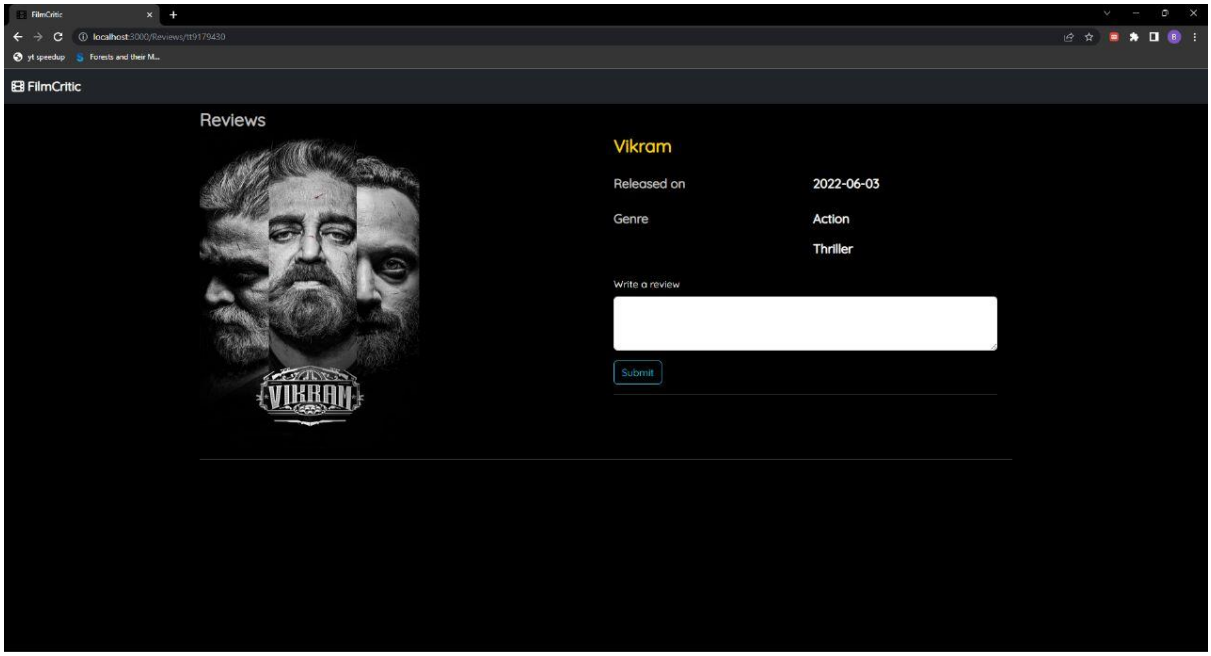


Fig 2: Movie details and Reviews section

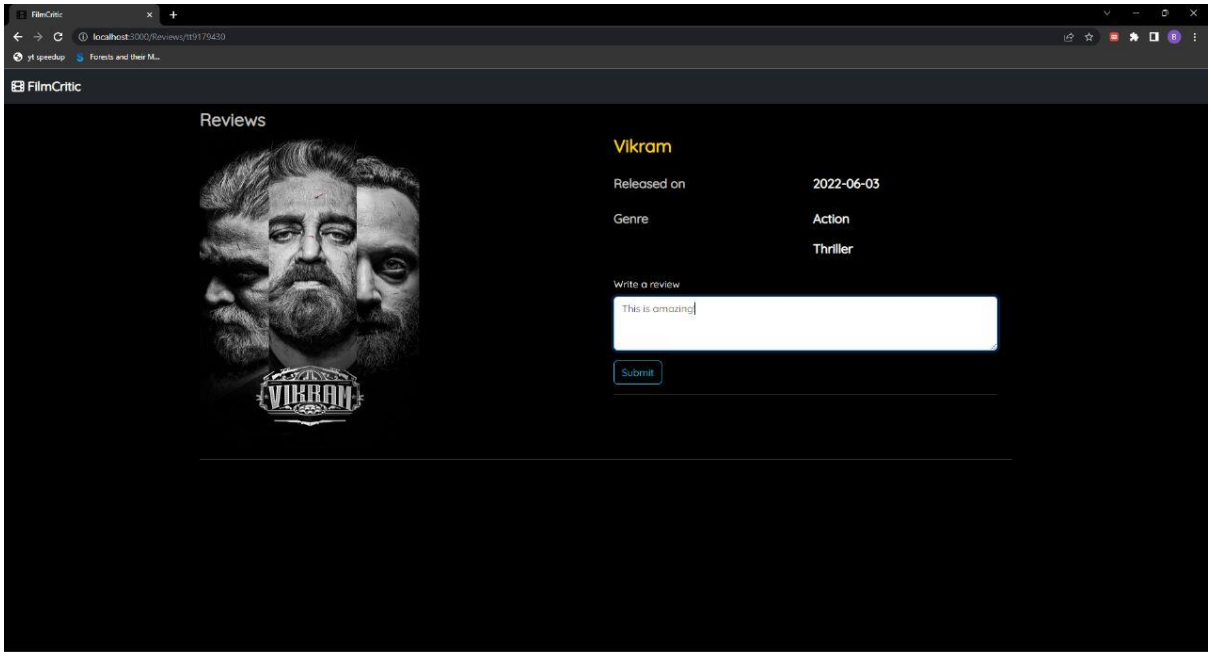


Fig 3: Providing review

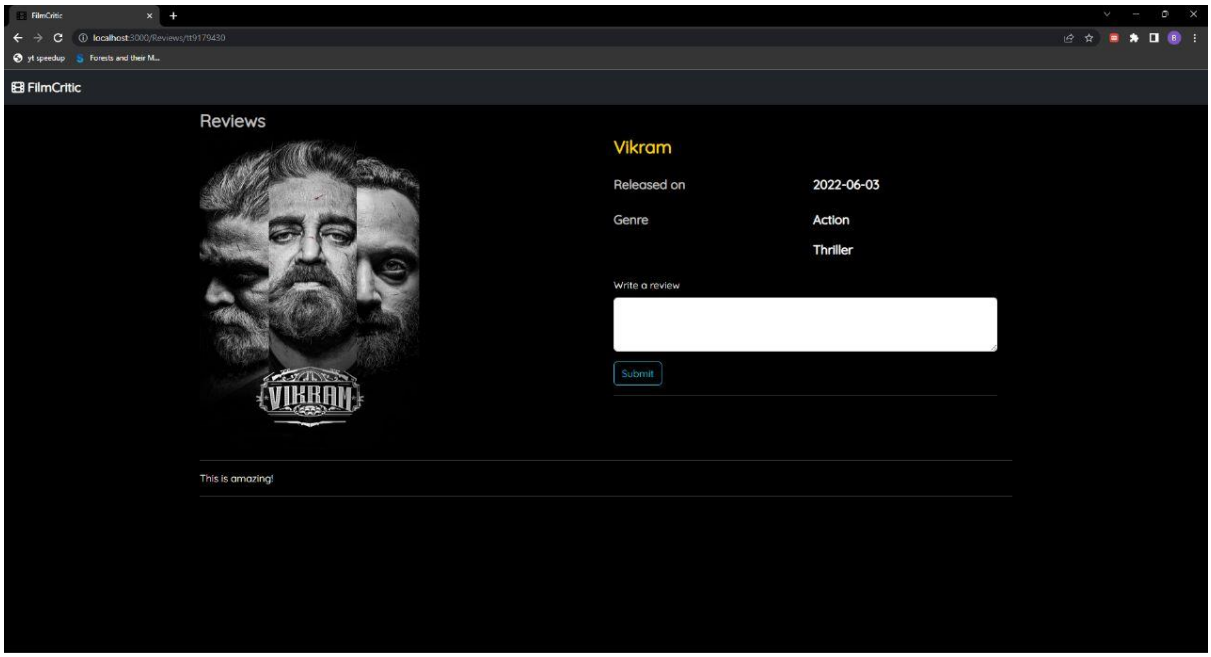


Fig 4: Review

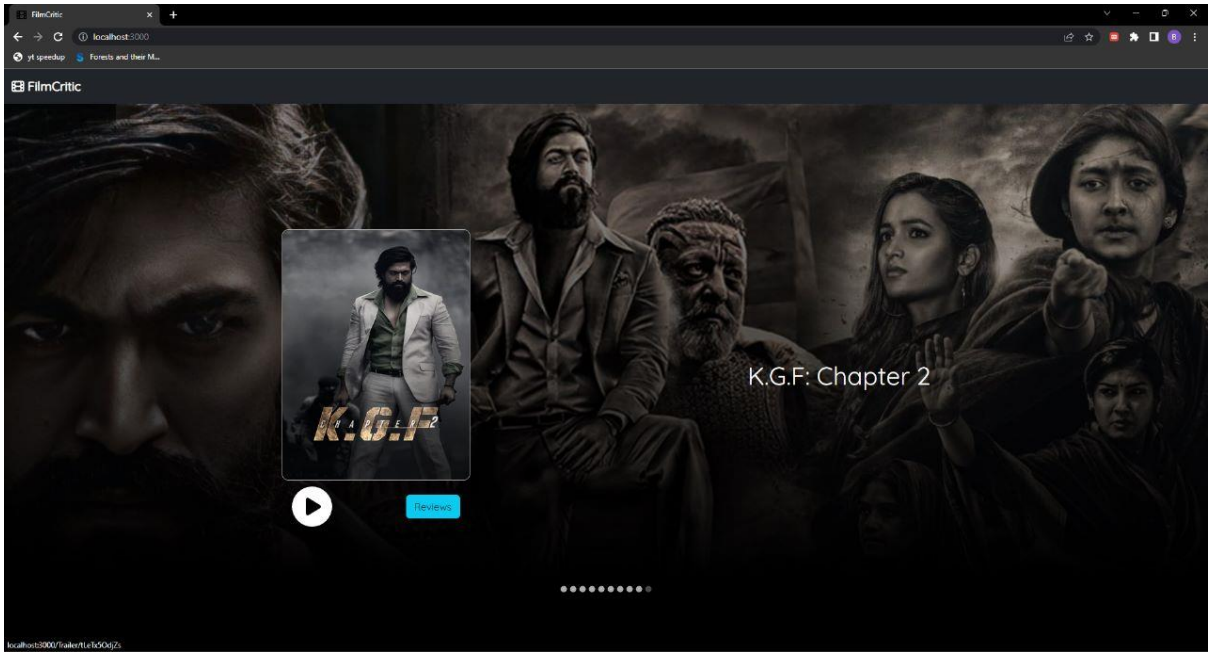


Fig 5: Home page

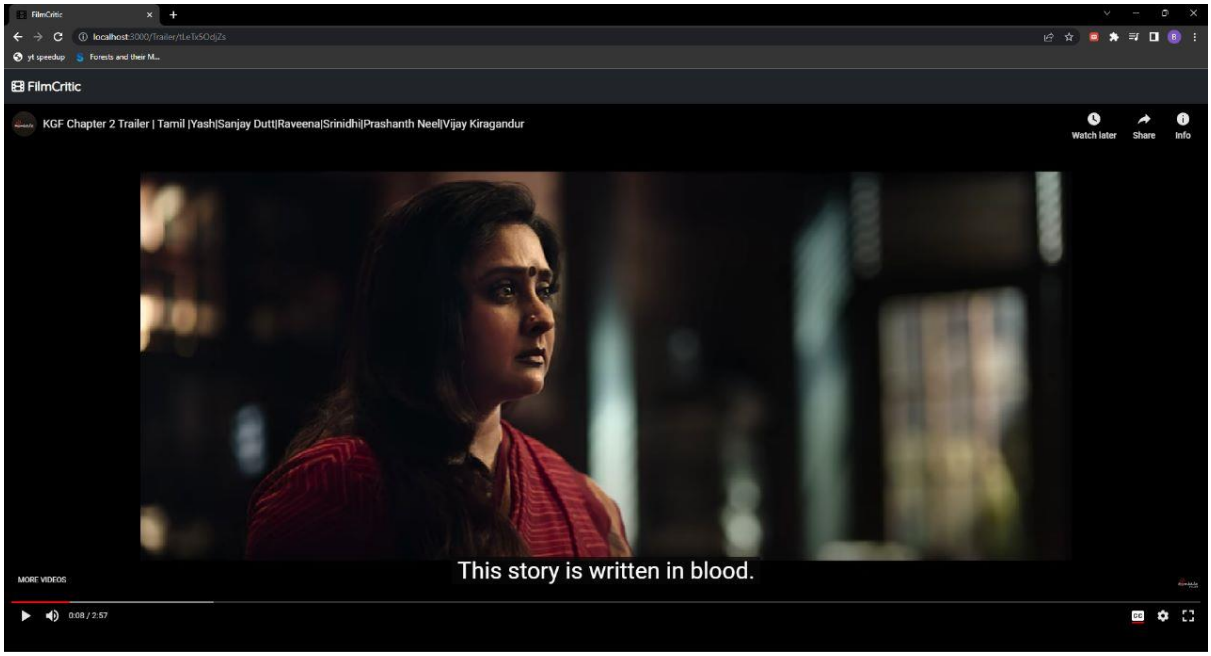


Fig 6: Playing trailer

7. ADVANTAGES & DISADVANTAGES

Advantages	Disadvantages
Guidance for decision-making	Subjectivity and bias
Expert opinions	Spoilers
Variety of perspectives	Limited scope
Discovering hidden gems	Misaligned expectations

Benefits of film reviews

- Saving time and money: By reading movie reviews, viewers can avoid spending their time and money on films that have gotten mixed or negative reviews.
- Discussions are sparked by movie reviews, allowing viewers to interact with others who have watched the movie and express their own ideas.

Demerit of Movie Reviews:

- Influence & Preconceived Ideas: Before a spectator even sees a movie, movie reviews can significantly impact their thoughts and expectations.
- Lack of own Connection: Reviews might not take into account a person's particular likes, preferences, or cultural upbringing, which causes a chasm between the reviewer's judgment and the audience member's own delight.

8. APPLICATIONS

Movie reviews are used in areas such as film journalism, academic research, film festivals, awards, streaming platforms, and film marketing.

The purpose of most movie reviews is to help readers determine if they want to watch, rent, or buy the movie. The review should give enough details about the movie so that the reader can make an informed decision, without giving anyway any essentials such as the plot or any surprises.

Movie critics act as advisors to consumers telling them which movies will be worth their money. The IMDb app has been specifically developed as a source of entertainment for millions of people around the world.

Movie review applications can be applied in the following areas:

- Consumer decision-making
- Film promotion and marketing
- Community engagement
- Film criticism and analysis
- Personalized recommendations
- Film industry insights

9. CONCLUSION

In conclusion, there are several benefits to using movie review software apps for watching films and making decisions. These apps offer easy access to a variety of movie reviews, enabling users to choose the movies they want to watch with knowledge.

Review subjectivity and prejudices, as well as the possibility of spoilers, should, however, be taken into consideration. Despite these issues, movie review software programs are useful resources for movie fans, filmmakers, and the entire film business.

It encourages participation in the community by enabling users to connect with others who share their interests and take part in discussions. The program aims to enhance the movie-watching experience through the dissemination of insightful information and the facilitation of well-informed judgments in an approachable and user-friendly manner.

10. FUTURE SCOPE

In the future, movie review software can be improved by adding sophisticated recommendation algorithms, integrating with social media for simple sharing and discussion, encouraging user-generated content, offering real-time updates and notifications, fostering community engagement through forums and chat rooms, integrating with streaming services for a seamless movie-watching experience, adding gamification elements, making sure accessibility features are present, and utilizing these technologies.

These improvements are intended to give users of the movie review software a more tailored, interesting, and inclusive experience.

1. It can be further extended for generating reviews related to the products in Online Shops, and e-commerce websites.
2. It can be used for generating reviews for online videos, advertisements, etc.
3. It can also be used for generating reviews related to the colleges during the admission process.

4. It could be enhanced for use in generating reviews of the candidates in the election.

There is a possibility to implement a search functionality into this project which makes it easier to search various movies into the database.

11. BIBILOGRAPHY

[1] Harikrishna V Holla proposed a project Web Development with ReactJS and Spring Boot on July 2020. 10.17148/IJARCCE.2022.11781

[2] Kumar, Manoj & Yadav, Dharmendra & Singh, Ankur & Kr, Vijay. (2015). A Movie Recommender System: MOVREC. International Journal of Computer Applications. 124. 7-11. 10.5120/ijca2015904111.

[3] Chao-Chun Chen; Min-Hsiung Hung; Kuan-Chou Lai; Yu-Chuan Lin, "Docker and Kubernetes," in Industry 4.1 (2022) : Intelligent Manufacturing with Zero Defects , IEEE, 2022, pp.169-213, doi: 10.1002/9781119739920.ch5.

[4] M. Gajewski and W. Zabierowski, "Analysis and Comparison of the Spring Framework and Play Framework Performance, Used to Create Web Applications in Java," 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2019, pp. 170-173, doi: 10.1109/MEMSTECH.2019.8817390

[5] Guntupally, Kavya, Devarakonda, Ranjeet, and Kehoe, Kenneth E. Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example. United States: N. p., 2018. Web. doi:10.1109/BigData.2018.8621924.

[6] Dartmouth Writing Program Handout on Film:

http://www.dartmouth.edu/~writing/materials/student/humanities/film.shtml

This handout is more focused on writing about film than on visual literacy, and it discusses different approaches to film (film history, ideological analysis, cultural studies/national cinemas, and auteur theory) not addressed in the Duke Writing Studio handout. It also features a short glossary of film terms.

[7] Internet Movie Database (IMDb): www.imdb.com

For quick information about a film, director, actor, producer, or production company, IMDb can't be beaten. It is not an ideal place to end your research, but it is a fine place to start.

APPENDIX

A. Source Code

Github link: https://github.com/Bhadhri03/FilmCritic