

International Journal of Research Publication and Reviews

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

Movie Times: A Movie Search Web Application

Suraj Mane ^a, Atharva Mhatre ^a, Sahil Kule ^a, Prof. Pallavi Marulkar ^b*

^a Students, Department of Computer Engineering

^b Professor, Department of Computer Engineering

ABSTRACT

In today's era of digital entertainment, users often search for movies online across multiple platforms to find information related to their favorite films, actors, or genres. However, most existing platforms focus only on providing movie details without offering a personalized space to maintain or organize a list of favorite movies. Managing and tracking personal watchlists becomes difficult when users have to rely on external tools or note them manually. To overcome this limitation, this paper presents "MovieTimes," a responsive and dynamic web-based application developed using React.js.This application enables users to search for movies using The OMDB API (Open Movie Database API) API, which provides accurate and real-time movie data such as movie names, posters, ratings, release dates, and overviewsMoreover, users have the flexibility to add or remove movies from their personalized favorites list for easy access later. These selected movies are safely stored in the browser's local storage, eliminating the need for server-side databases while ensuring data remains available. The primary goal of this project is to provide a fast, efficient, and user-friendly movie search experience with low technical complexity. The application features a clean, intuitive interface, responsive design for various devices, and helps users manage their movie preferences in an organized way.

Keywords: Movie Search Web App, OMDB API, MovieTimes, React.js, Favorite Movies, Movie Watchlist

1. Introduction

With the growing popularity of digital entertainment platforms, searching for movies online has become a common practice among users of all age groups. Although several movie search websites and applications provide detailed movie information, only a limited number offer users the added convenience of creating and managing their own personalized list of favorite movies within the platform itself. Users often need to depend on external tools or manual methods to keep track of the movies they like or wish to watch later. MovieTimes addresses this limitation by offering an integrated solution where users can effortlessly search for movies and save them directly to their personalized favorites section within the same application environment. This eliminates the need for any third-party services for managing movie preferences. The application has been developed using React.js, a popular JavaScript library for building dynamic user interfaces. It utilizes The OMDB API (Open Movie Database API) API to fetch real-time movie data, ensuring that users receive the most updated information. The platform displays essential movie details such as posters, ratings, release dates, and short descriptions in an organized and user-friendly layout. Additionally, a dedicated favorite section enables users to view, add, or remove movies from their watchlist, thereby enhancing the overall movie search experience in a simple yet efficient manner.

1.1 Problem Statement

In today's digital world, finding movie information across multiple platforms can be time-consuming and inconvenient. Existing movie platforms provide extensive data but lack simplicity, real-time search efficiency, and favorite management within a lightweight system. To overcome these limitations, "MovieTimes" is developed to allow users to search for movies easily, view important movie details, and create their personalized favorite movie list. The application aims to provide a minimal, responsive, and user-friendly interface for enhanced user experience.

1.2 Methodology

The development of the *MovieTimes* movie search web application follows a structured and well-organized methodology to ensure simplicity, userfriendliness, and efficient performance. The development process started with requirement analysis, focusing on identifying key user needs such as movie search functionality, viewing essential details, and managing a personalized favorites list within a single platform.

The system utilizes a client-side architecture, with React.js employed for building an interactive and responsive front-end interface. To fetch real-time movie information based on user input, the OMDB API (Open Movie Database API) is integrated. This API provides essential movie data, including titles, release years, ratings, posters, and brief descriptions, ensuring accurate and updated content.

The application follows a component-based architecture using React.js, promoting modular development, reusability, and faster rendering. The system is divided into key modules such as the Search Module, Movie Display Module, and Favorite Management Module. User-selected favorite movies are stored locally in the browser's storage, allowing persistent data access without relying on a backend server.

The search functionality enables users to look for movies using keywords, retrieving results from the OMDB API and displaying them dynamically. Each movie result is presented in a card layout containing relevant details along with options to add or remove it from the favorites list. The development approach followed the agile methodology, ensuring iterative progress, continuous testing, and incorporation of user feedback. Functional testing was conducted to validate key features like search operations, movie detail display, favorites management, and responsiveness across various devices.

The project was deployed using Netlify, providing a secure, optimized, and scalable platform for hosting the application. Future improvements for the MovieTimes project may involve integrating user authentication, cloud storage for syncing favorite movies across devices, advanced search filters, and personalized movie recommendations based on user preferences. This methodology ensures a smooth, user-friendly, and engaging movie search experience while maintaining simplicity and efficiency in design.

1.3 Objectives

To develop a user-friendly movie search platform that allows users to search for movies using keywords and view relevant information like title, rating, release year, poster, and description.

To integrate the OMDB API (Open Movie Database API) for fetching real-time movie data and ensuring accurate and up-to-date movie information.

To implement a favorite movie management system where users can add or remove movies from their favorite list based on their preferences.

To store and manage favorite movies using the local storage feature of web browsers, allowing persistent data storage without the requirement of a backend database.

1.4 Requirement Specification

Table 1: Software Requirements

OPERATING SYSTEM	WINDOWS OS/ ANY OS
IDE	VISUAL STUDIO CODE
SOFTWARES	REACT.JS, FIREBASE, NODE.JS, VERCEL

Table 1: Hardware Requirements

СРИ	MINIMUM 2 CORES AND 4 THREADS
RAM	MINIMUM 4 GB
MEMORY	MINIMUM 128 GB

2. System Architecture

The architecture of MovieTimes is based on a simple client-side design using React.js. The user interacts with the web interface to search for movies. The application sends requests to the OMDB API to fetch real-time movie data like title, poster, rating, and release year. The fetched movie details are displayed on the screen in a card layout. Additionally, the system provides a favorite section where users can add or remove their selected movies. These favorite movies are stored securely in the browser's local storage, allowing users to access them anytime without the need for an internet connection or backend server. This architecture ensures a lightweight, fast, and user-friendly movie search experience.



Fig. 1 - (a) System Architecture

2.1 Conclusion

The MovieTimes web application provides a simple and efficient platform for users to search for movies and manage their favorite movie list in a userfriendly environment. Developed using React.js and integrated with the OMDB API, the system offers real-time access to accurate movie information such as titles, ratings, release dates, and posters. The application successfully meets the objective of allowing users to create a personalized favorite movie list without the need for complex backend storage. By utilizing the browser's local storage, users can easily add or remove movies from their favorites and access them anytime. The lightweight design, responsive user interface, and fast search functionality make MovieTimes a reliable and convenient platform for movie enthusiasts. The system architecture ensures smooth navigation and quick access to information, enhancing the overall user experience. In the future, the application can be improved by adding features like user authentication, cloud-based storage for cross-device favorite management, movie recommendations based on user preferences, and integration with streaming platforms to provide a complete entertainment solution.

Results



Fig. 2 - (a): Home Page

References

John Williams, Emily Clark (2024). MovieBase: A Web Application for Real-Time Movie Search and User-Friendly Favorites Management.. Biman Saha (2024). Development of Movie Search Applications Using React.js and RESTful APIs: A User-Centric Approach. Rahul Sharma, Sneha Patel (2025). Optimizing User Experience in Entertainment Applications Using Local Storage and Client-Side Data Handling. OMDB API Documentation (2024). Open Movie Database API for Fetching Real-Time Movie Data. Available at: https://www.omdbapi.com/ Miguel Ángel García-Madurga, Ana-Julia Grilló-Méndez (2023). Artificial Intelligence in the Entertainment Industry: An Overview of Reviews. W3Schools(2024).Modern Web Development: Local Storage Implementation in HTML5