MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

VI Semester B.Tech. (IT)

ICT 3266: INTERNET TOOLS AND TECHNOLOGY LAB

MINI-PROJECT IMPLEMENTATION DOCUMENT

MOVIE REVIEW SITE

A PROJECT REPORT

submitted by

TEAM ROUTERS

SANKET KAR Reg number: 200911264

RAVI SHARMA Reg number: 200911268

ABSTRACT

The aim of this project is to design and develop a comprehensive web application that provides users with a vast database of movies and TV shows. The application will allow users to browse and search for information on various titles, including their ratings, reviews, cast, crew, trailers, and more. The application will also provide a comprehensive search functionality that will allow users to search for movies and TV shows based on criteria such as title, genre,etc. Users will be able to create their profiles, rate and review movies and TV shows, and interact with other users through a commenting system. The project will involve designing a robust database schema, implementing data modeling, developing APIs, and building a user-friendly interface. The ultimate goal is to create a highly scalable and user-focused platform that provides users with a rich and immersive experience when browsing and exploring movies and TV shows. The proposed project will be developed using a full-stack technology stack, comprising ReactJS for the front-end, Express.js for the server-side framework, and MySQL for the database management system.

INTRODUCTION

The framework used to build the front of the website is ReactJS. An extensive number of libraries have been integrated to bring in some production-level standards.

This includes usage of design components from Bootstrap and Antd.

SkeletonJS is used for Skeleton loading effect. Skeleton loading is a technique that shows temporary placeholders resembling the content being loaded, providing users with a sense of progress and enhancing the overall experience while waiting for the actual content to load. Thereby, page loading is seemless and clean to the eye with minimum time delay.

Extensive JSX and CSS has been used to create a beautiful-looking and friendly user-interface.

The backend used is ExpressJS on NodeJS. This is the interface between all interactions from the frontend and the database. The backend consists of creation of a large number of APIs which the frontend utilizes to fetch data while maintaining user authentication and authorization. This filters data and organizes them to only the fields of purporse and maintains invisibility of the frontend to entire control of the database.

Extensive production standard libraries have been used for the backend such as:

BCryptJS used for password hashing/encryption, JWTokens to track logged in user, SkeletonJS for skeleton loading, Cors for restricted http requests, and Express server to host the APIs.

The project will have the following components: A login/sign up page (avoidable just to scroll through movies but not review), integration with TMDB API for popular, now-playing and upcoming movie listing, category wise listing, an information filled page about each movie, comment box to review and rate the movie, global rating of the movie on an algorithm from all user ratings, about us page. The NodeJS backend is connected to a MySQL database which is the technology used for the backend to store, maintain, and update our database.

OBJECTIVES

The objectives of the website are to:

➔ To develop a community between users.

➔ To give detailed information about a movie.

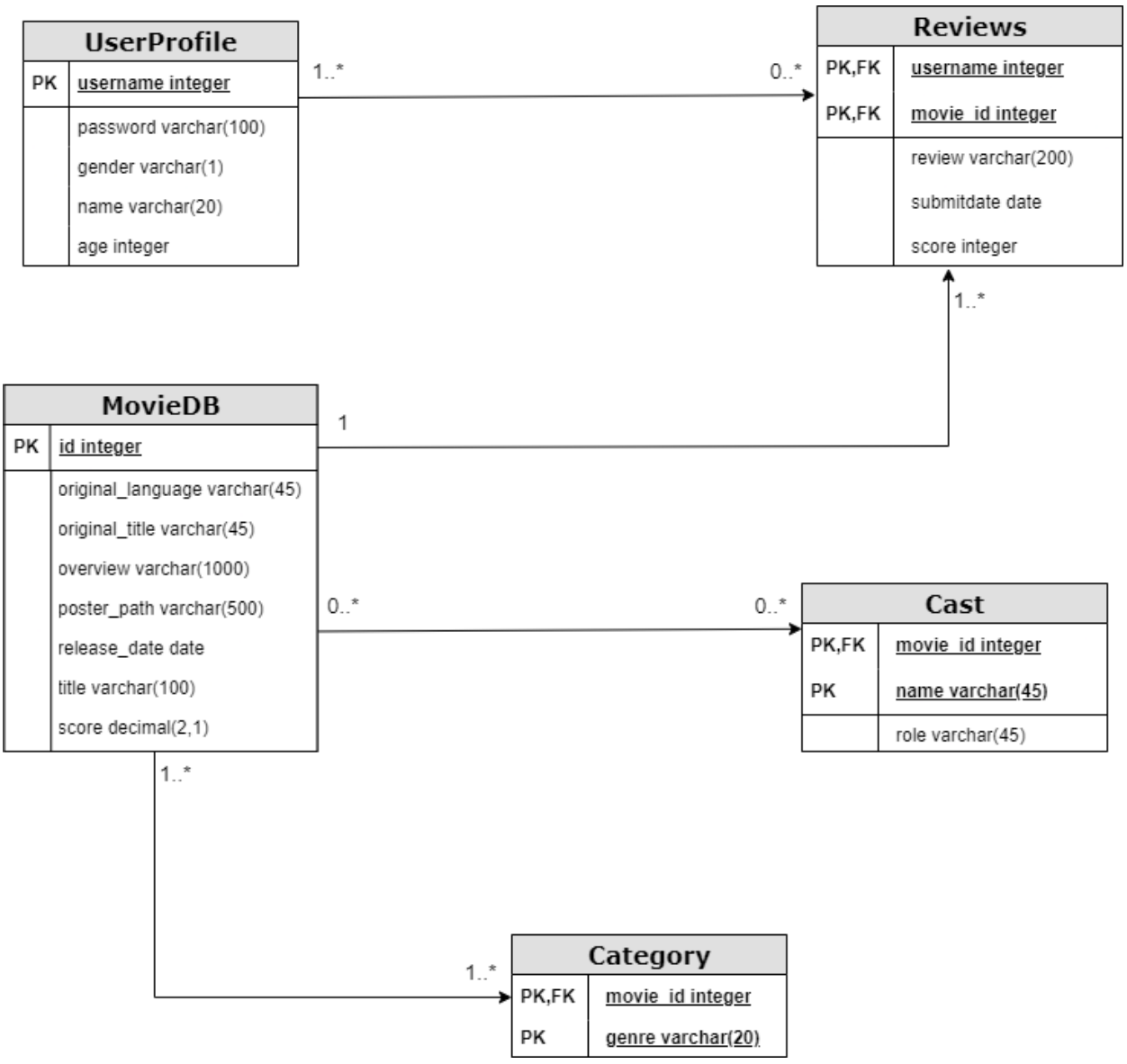
➔ To find movies based on genre filters

➔ To get movie listing details on currently playing and upcoming movies

➔ To judge movie ticket purchase based on community feedback and ratings.

➔ Having clean interface to rate and review movies.

SCHEMA DIAGRAM



SCHEMA IMPLEMENTATION

UserProfile (age int not null, gender varchar(1) not null, name varchar(20) not null, username varchar(20) not null, password varchar(100) not null, primary key (username))

moviedb (id int not null, original_language varchar(45) not null, original_title varchar(45) not null, overview varchar(1000) not null, poster_path varchar(500) not null, release_date date not null, title varchar(100) not null, score decimal(2,1) not null, primary key (id), unique key id_unique (id))

review (username varchar(20) not null, review varchar(200) not null, submitdate date not null,
movie_id int not null, score int not null, primary key (username,movie_id), foreign key (username) references userprofile (username), foreign key (movie_id) references moviedb (id))

category (movie_id int not null, genre varchar(20) not null, primary key (movie_id,genre), foreign key (movie_id) references moviedb (id))

cast (movie_id int not null, name varchar(45) not null, role varchar(45) not null, primary key (movie_id,name), foreign key (movie_id) references moviedb (id))

TECHNOLOGIES USED

- HTML/JSX
- CSS
- NodeJS
- JavaScript
- AntDesign
- ReactJs
- TMDB API
- SkeletonJs

IMPLEMENTATION

Movie Review System functionally works on a separately hosted front end and back end system. In the back-end, NodeJS functions have been built to query MySQL while developing them into functional and tested APIs. These APIs control data access between what is present in the database v/s what is visible to the various processes in the front-end. The front-end retrieves data from the back-end via the API calls and has no direct access with the database. This allows data-protection from possible vulnerabilities.

User-Interactive Features:

- Category-wise movie listing - The category wise movie listing feature in a movie review system allows users to browse movies by genre or theme. Users can select a category to view a list of movies belonging to that category, and search for movies based on keywords, actors, or directors. This feature enhances the user experience and increases engagement.

- Real-Time movie details – Using the TMDB API, the user has access to curated information about movies that are currently in theatres and those that are upcoming. This increases the activity of the website as it is a great incentive for users to visit the website.

- Detailed Information about a movie – Users are provided with a detailed list of details about a movie. All details provided are interactive. For example, each cast member is hyperlinked with an image. Each production company mention links to their website. This fulfils a user's curiosity without him needing to make the needful google searches by himself. The site is self-sufficient with information delivery.

- Ability to read others' reviews - Users can view reviews and ratings of movies left by other users, helping them make informed decisions about which movies to watch. This feature encourages community engagement and fosters a culture of sharing opinions and insights about movies.

- Create your own reviews and ratings - The create your own reviews and ratings feature allows users to write reviews and ratings of movies they have watched, encouraging user engagement and sharing of opinions.

Security Features:

- Password hashing – The library we used for password hashing is 'node.bcrypt.js'. Bcrypt is a widely-adopted password hashing algorithm that is fast and uses a unique salt for each password, making it more difficult to crack through

brute-force attacks. Bcrypt is also an adaptive hash function, allowing it to adjust to computing resources and is widely used and tested in many systems.

- Authentication Tokens - JWT (JSON Web Token) is a compact and self-contained way of transmitting information securely between parties. It's commonly used for authentication and authorization in web applications, containing a set of claims that are encoded and digitally signed with a secret key. Once a user is authenticated, a JWT token is sent to the client and subsequently used for every request to the server, allowing secure access to resources based on the user's permissions. This way repeated signing in was avoided, it was possible for the front end session to understand which user was logged in, and delivery of information only when authenticated.

- Input Validation - Input validation is the process of checking and validating data input to an application, such as user input from forms or API requests. Proper input validation helps prevent malicious code injection, data breaches, and other security vulnerabilities by ensuring that input data conforms to the expected format and is safe for processing. Our website had quite a lot of input from users, and everywhere was validation maintained. This also gave back the user true feedback on failed submission, and security breach avoidance.

DATABASE USED

The database we chose to use is MySQL. MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements. MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.. MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

EXPECTED RESULTS

Increased user engagement and participation: The website's interactive design and user-generated content will encourage users to participate actively by analysing, rating, and reviewing campgrounds.

Increased traffic and user retention: The website's user-friendly interface and loading, comprehensive and live updated movie list, and search by genre feature are expected to attract more users and keep them engaged.

Improved user experience: The website's responsive design, rating system, and user-friendly interface are expected to improve the user experience and provide a seamless platform for movie enthusiasts.

Usage of ML/AI: Data can be tracked on every user’s activity and likings on the website based on his viewing and rating patterns. This can be ultimately used to have a highly accurate movie suggestion algorithm.

CONCLUSION

In conclusion, the development of the Movie Review System is a great example of how web technologies and database management skills can be utilized to create a valuable and user-friendly application. The system provides a platform for movie enthusiasts to browse, search, and rate movies, as well as read reviews from other users.

The project showcases the ability to design and implement a responsive and visually appealing user interface using HTML, CSS, and JavaScript. The use of MySQL for backend development and database management further enhances the functionality and reliability of the system.

Overall, the Movie Review System project is a great example of web development and database management skills and demonstrates the ability to create a functional and user-friendly web-based application.