CriticsWorld

DEVELOPER GUIDE

Version 1.0

- Sindooja Bhagwan Gajam (2888228)
- Praveen Sai Kodali (2883944)
- Manoj Sai Yerramaneni (2885415)

1.0 Introduction and Background

1.1 Introduction

CriticsWorld is a platform deliberately designed to change how people interact with the world of movies. It is a web application where users can post their reviews about the movies they have watched or get an opinion on whether to watch a movie based on the reviews posted.

Criticsworld can be used with 2 roles: users, and admin. Criticsworld provides a unified platform for users to post, read, edit, and delete reviews. On the other hand, the admin has the same functionalities as the user, but the admin is solely responsible for adding, updating, and deleting movies from the application.

The purpose of this developer guide is to provide the maintainer with important information on architecture, design, maintenance guide, and suggestions for future improvement of the software. If at any time this guide does not contain the information that the maintainer needs, the reader can use any of the main deliverables of Criticsworld that provide more granular information about the software design and implementation.

1.2 Objectives

Criticsworld provides the following functionalities:

- User functionalities include:
  - Registration
  - Login
  - Reset Password
  - Post Review
  - Edit Review

- Delete Review
- Search Movie
- Search Review
- Edit Profile
- Delete Profile

- Admin functionalities include:
  - Login
  - Reset Password
  - Add Movie
  - Update Movie
  - Delete Movie
  - Search Movie
  - Post Review
  - Edit Review
  - Delete Review
  - Edit Profile
  - Delete Profile

2.0 Architecture

2.1 Software Architecture

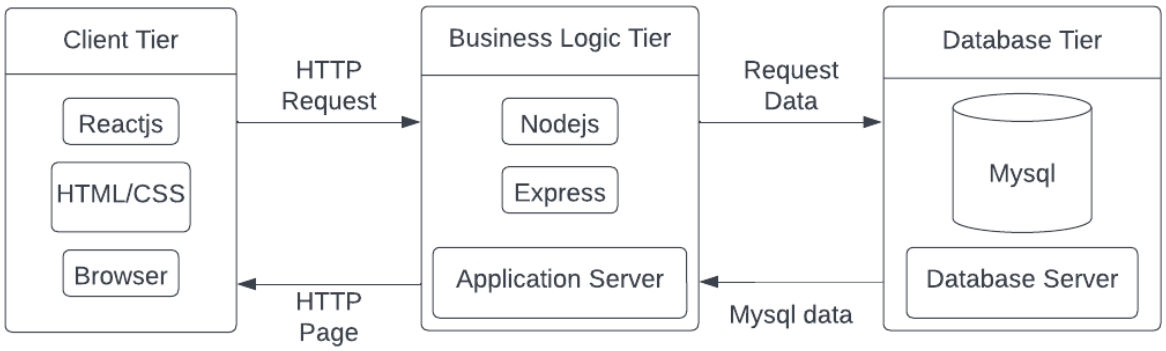

Figure 1 Architecture Diagram

This design shows how the request flows through the system. The request is made from the client tier to the business layer tier and in the business layer tier the request is formatted according to the database understanding and then the request is made to the database to get a response. Once the application server receives the response it is formatted as per the HTML and sent it back to the client tier.

2.2 Software Interfaces

Criticsworld uses several off-the-shelf software packages to achieve its goals. When you're setting up Criticsworld, you'll need to be sure (in most instances) that you've got the following software installed on your hosting system:

- Visual Studio Code: Official Microsoft IDE for JavaScript. It is widely used by JavaScript developers. We have used VS code version 1.82.1 for our project. It is recommended that the maintainer utilize at least this version of the software.

- MySQL: It is recommended that the maintainer have MySQL 8.0.35 version or above installed.
- NodeJS: Node.js is an open-source JavaScript runtime environment that is cross-platform, and capable of operating on Windows, Linux, Unix, macOS, and other platforms. We have developed the Criticsworld web application on node v20.10.0. It is recommended to have at least this or the above version to be installed.
- ReactJS: React.js is a JavaScript library for building user interfaces, particularly for single-page applications where UI updates are frequent. Developed and maintained by Facebook, react is widely used for its efficiency in creating interactive and dynamic user interfaces. We have used React version 18.2.0 for Criticsword. We recommend the same or above version for maintenance.

3.0 Database Design

Criticsworld follows a relational database model. The data is checked for correctness before it is entered into the database anyway, so no invalid values can exist in the affected tables.

Criticsworld's database design schema is listed below:

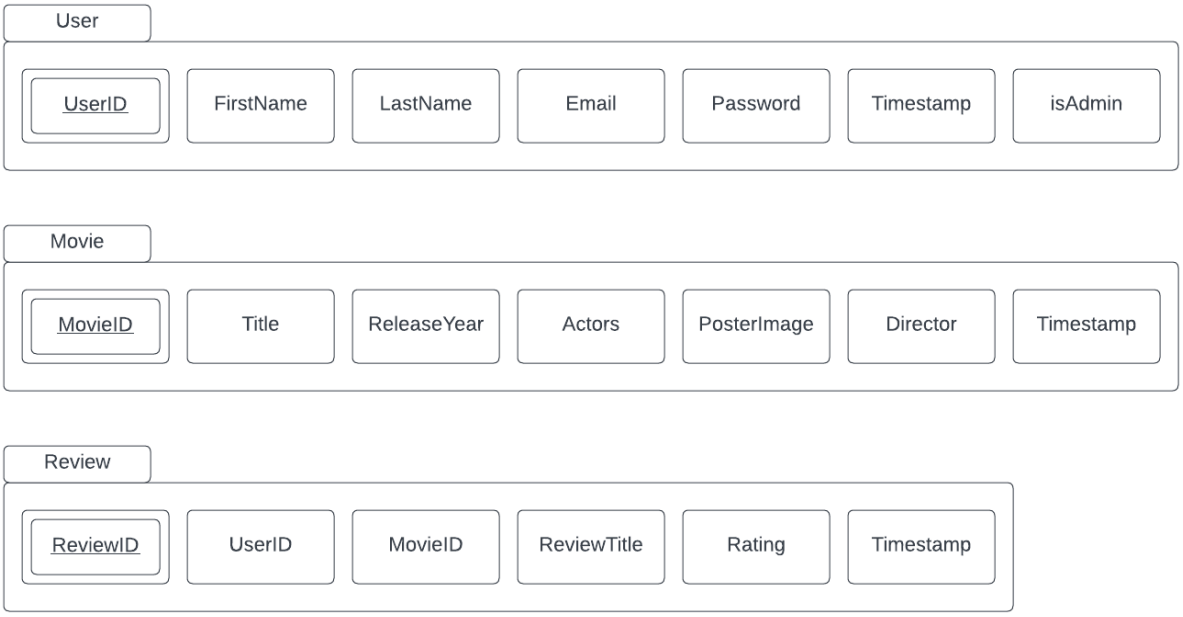

Figure 2 Database Design

Following is a description of the tables presented in the ERD:

Description of the Tables Present in the Criticsworld ERD	
Table Name	**Description**
User	This table holds the records related to registered users of the web application.
Movie	This table holds the records of the movies, and this is one of the primary tables in the database model.
Review	This table holds the records of the reviews posted on the movies. The UserID, and MovieID are the foreign keys.

Following is a presentation of Criticsworld's data dictionary:

Data Dictionary for Criticsworld		
Name	**Data Type**	**Description**
MovieID	Integer	Uniquely identifies movie in a system. This is also the foreign key in the Review table
Title	String	A title of movie. Constrained to 255 characters
ReleaseYear	Integer	Holds the year when the movie was released
Actors	String	Holds the name of the actors in the movie

PosterImage	String	Holds the poster image data of the movie
Director	String	Holds director name of the movie. Constrained to 255 characters
ReviewID	Integer	Uniquely identifies review in a system.
ReviewTitle	String	Holds the review of the movie. Constrained to 255 characters
Rating	Integer	Holds the rating of the movie
UserID	Integer	Uniquely identifies user in the system.
FirstName	String	Holds user's first name. Constrained to 255 characters
LastName	String	Holds user's last name. Constrained to 255 characters
Email	String	Holds the email id of the registered user. Constrained to 255 characters
Password	String	Holds the password of the user
Timestamp	timestamp	Holds the current timestamp.

3.0.1 User

This is the most crucial class used by Criticsworld. This table contains information about users like their name, email ID, role, and various other details. The user has two different kinds of roles to access the system with some different sets of functionalities. User is logged in to the system whenever they log in and deactivated whenever they log out. This stops a user from having multiple active sessions. It also stops a user from manipulating client-side web controls to view pages that their role does not allow them to view. The user can edit or delete the profile. The profile once deleted can never be retrieved. The admin role can do movie crud operations and review crud operations while the non-admin role user can only do review crud operations.

3.0.2 Movie

This table contains information about the movie like title, actors, director, and various other details. Only the user logged in with an admin role will have the ability to add, update, or delete the movie from the system. A user once logged in to the application the application will redirect the user to the dashboard which will show the list of movies from the database. Users can select any movie and it will navigate to the movie details page, which will show the information about the movie.

3.0.3 Review

This table contains information about the reviews posted on the movie, details like reviews, and ratings. A user will have the ability to add, edit,

and delete the reviews posted by themselves. Review once deleted cannot be retrieved ever.

4.0 Backend files/folders

4.0.1 package.json file

The package.json file contains descriptive and functional metadata about a project, such as a name, version, and dependencies. The file provides the npm package manager with various information to help identify the project and handle dependencies.

4.0.2 app.js file

The app.js file is the main entry point or starting script for the application. It's a convention to name the main file that kicks off your Node.js application as app.js, but you can choose any other name as well. This file is responsible for setting up the server, configuring middleware, defining routes, and starting the server. This file includes middleware configurations, additional route definitions, and another setup specific to the project.

4.0.3 Routes folder

This folder contains files to organize and modularize the route-handling logic of the application. We created separate folders and files for the routes to make the codebase modular and facilitate easier maintenance and scalability as the application grows.

5.0 Frontend files/folders

5.0.1 main.jsx file

The main.jsx file serves as the main entry point or a key file in our project. This file is often responsible for rendering the main component or setting up the React application.

5.0.2 index.html file

The index.html file serves as the entry point for our web application. It is a static HTML file where our React components are be injected.

5.0.3 App.jsx

The App.jsx file represents the main component of our application. This component is responsible for rendering the structure of our application and it handles the routing for our frontend.

6.0 Future Enchancement & Improvements

For the web application criticsworld, future improvements could be focused on enhancing the user experience, adding new features, and keeping the platform up-to-date. Here are some of the possible improvements:

- Personalized recommendation: Use algorithms to analyze reviews and preferences to provide personalized movie recommendations according to the user's history.
- User comments: Allow users to comment on each other's reviews to create a community and encourage interaction.
- Integration with external databases: Use external databases to collect additional information about movies, including genre, crew details, trivia, and trailer links.
- Mobile compatibility: Create a mobile app to make the platform available on mobile and tablet devices.
- Notification system: Use a notification system to notify users when there is a new review, comment, or discussion related to their favorite genre or movie.

7.0 Conclusion

I have tried to be as thorough as possible with this manual under rigid time constraints. Please be aware that this manual is for version 1.0 of the Criticsworld software and should be updated accordingly for future release versions.