Project Title	**Online Movie Rating/Review/Track System**
Technologies	MERN
Domain	Entertainment
Project Level	Medium

Table

Table of Contents

1. Problem Statement:

Design a web application “Online Movie Rating/Review/Track System” to allow users to browse movies and keep track of the movies they have watched and add a rating and review to the movie.

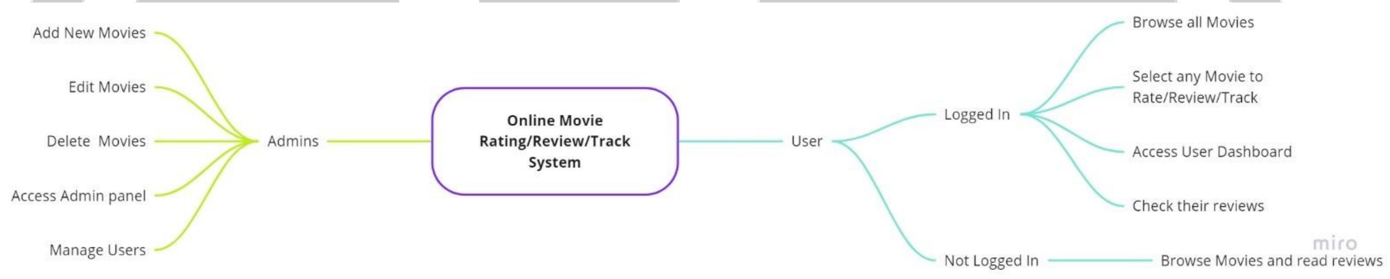
1.1. What is a Movie Rating/Review/Track System?

A Movie Rating/Review/Track system is an online tool which will have many movies listed and the users can come to the site browse the movies and if they have watched any movie, they can leave a review and can also rate the movie. They will also have an option to mark the movie as watched to keep track of the movies that they have watched. All these operations can only be performed by the users who are logged in.

You might have used Trakt to rate/review/track the list of movies and T.V. shows that you have watched.

Now that you understand what a Movie Rating/Review/Track System is, let’s discuss some of the functionality of the “**Online Movie Rating/Review/Track System**” that you will design.

1. To create a web-based system that will enable users to browse any movie available from the list along with the movie details.
2. Users must be able to search for the movies using a search box and relevant results should be displayed.
3. Users need to register themselves first before rating any movie, writing a review or marking movie as watched.
4. There has to be a user dashboard which should also show users their watched movies.
5. The admins can add/edit/delete any of the movies listed on the site.
6. All non-logged-in users must be able to browse the movies and see the information about the movies.
7. The project should be very easy to use with a clean UI.



Low-Level system flow

1.2. Project Objective

• To build a system where the users can come and give their views on any movie and other users can see the review and decide if the movie is worth a watch or not.

• To make the users life easy by giving them an option to mark any movie as watched so that they can check back in the future if they have seen that movie already or not.

1.3. Scope of The Project

1. The online bus booking application maintains the data of all the movies along with the movie information like the release date, main actors and actresses, etc.,

2. To give users the option to set the movie as watched so they can keep on binge watching and not worry about re-watching a movie that they have previously watched.

3. To give the users a clean and simple UI so that they can navigate the site easily instead of getting buried under 'X' number of features that the user does not require.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

Admin:

- Can manage all the data.
- Can add Movies.
- Can edit/delete any movie listed on the website.

User:

- Users can browse the available movies and can write a review, give a rating and mark the movie as watched after registration.
- Users can view the details of all the available movies and read other users reviews and see the overall rating.

UI:

- There should be a dashboard for admins, and registered users.
- Frontend should be clean and minimal which shows the available movies along with their info.
- Register/Login forms and a search box for easier discovery of desired content.

1.4.2. Non-Functional Requirements

1. **Security:** The system should have a certain level of security such that not anybody can have access to sensitive information and the passwords should be properly encrypted.
2. **Robustness:** If the user's system crashes, a backup of the user data must be stored on remote database servers to enable recovery.
3. **Performance:** The application must be lightweight and the UI should be fast and responsive.
4. **Authentication Requirements:** There should be an authentication system in place which checks the credentials each time a user logs in.

1.5. Modules List

Admin:

- Login
- Add movies o Title o Description o Release Date o Actors, etc.,
- Update/Delete Movies **User:**

- Login/Register
- View Movies
- Review/Rate a movie
- Mark the movie as watched
- My Profile
- Logout

2. Project Evaluation metrics:

2.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Deploy the application on your preferred service.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You must submit a system architecture design in your wireframe and architecture documents.

2.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.