

Challenge **Movie Database Rest API Backend**

Problem Statement

To build an Rest API server like IMDB, where,

- public users can use the service to search for movies by name, genre and be able to sort and filter the results by votes, genre
- Registered/Authenticated users should be able to add movies, upvote/downVote movies, set favourite genre and get recommendations based on favourite genre.

Problem Description

Registration Services:

- **Register/Signup** - Api for user registration (username/password should be fine)
- **Login Api** - Registered user should be able to login and use other Apis

Services that can be used by registered users only:

- **Add Movies Api** - A registered user must be able to add a movie to the DB. A movie must have a name, genre, release date, up-votes, down-votes and reviews.
- **Set favourite genre Api** - A registered user must be able to set their favourite genre.
- **Get Recommendations Api** - A registered user must be able get a recommended list of movies based on their favourite genre they had set earlier.
- **Get Movie Detail Api** - A registered user must be able get all the details about a movie (name, genre, release date, up-votes, down-votes and reviews)
- **UpVote/DownVote Api** - A registered user must be able to upvote or down-vote a movie.
- **Add Review Api** - A registered user must be able to add a review for a movie.

Public Api:

- **Get Movies api -.**

Query all
movies

sort by upvotes/downVotes

sort by release
date

*** Add necessary validation for each endpoint and add tests where required.**

Expected Outcome

- A link to the original code base in a git repository of your choice.
- A very well documented ReadMe.md to build and run the application locally or in the cloud, detailing the framework / language and all third party frameworks used.
- If swagger documentation is not part of the code, A postman collection export needs to be attached.
- A link to deployment service such as Herokku, Netlify, AWS Amplify if the application is hosted
- Add unit test cases for the application