Summary

This challenge is designed to put your skills to the test by designing and building a good RESTful API to manage a small movie rental.

Time frame

Max 7 days.

Requirements

1.Only users with admin role are allowed to perform the following actions:
2.Add a movie
3.Modify a movie
4.Remove a movie
5.Delete a movie
6.Movies must have a title, description, at least one image, stock, rental price, sale price and availability.
7.Availability is a field of movies, which may only be modified by an admin role.
8.Save a log of the title, rental price and sale price updates for a movie.
9.Users can rent and buy a movie. For renting functionality you must keep track when the user have to return the movie and apply a monetary penalty if there is a delay.
10.Keep a log of all rentals and purchases (who bought, how many, when).
11.Users can like movies.
12.As an admin I’m able to see all movies and filtering by availability/unavailability.
13.As an user I’m able to see only the available movies for renting or buying.
14.The list must be sortable by title (default), and by popularity (likes).
15.The list must have pagination functionality.
16.Search through the movies by name.

Security requirements

1.Add login/logout functionality.**Preferably JWT.**
2.Only admins can add/modify/remove movies.
3.Only logged in users can rent and buy movies.
4.Only logged in users can like movies.
5.Everyone (authenticated or not) can get the list of movies.
6.Everyone (authenticated or not) can get the detail of a movie.
7.Publish your work using heroku and share the link with us.

Extra credit

1.Recovery and forgot password functionality (send email).

2.Confirming account (send email)
3.Build a small frontend app and connecting to the API.
4.As an user with admin role I want to be able to change the role of any user.
5.Unit test, at least 80% of coverage.
6.Include a dockerfile for production deployments.

Keep in mind

•You are free to use any package, library and weapons for the battle as long as you can justify their use.
•You may use any kind of database you like.
•Provide a database dump so we can replicate the database locally.
•POSTMAN will be used to evaluate the API. It would be great if you can provide us with a collection to test your API.
•Follow best RESTful API practices. Take a look on this guidehttps://www.vinaysahni.
•Use git and do small commits to facilitate the evaluation of progress.
•Include a **readme.md** file with instructions on how to setup your project locally to test it. (This is super important, if we cannot install it and run it easily we cannot evaluate it).
•Upload your solution to your Github or Gitlab accountand share a link with your evaluator.
•The test has been designed with enough time to do a good job, so don’t cut any corners, take your time and watch for quality. We evaluate code readability, comments, formatting, performance and re-usability.