**Sr Software Engineer Case Study (Full Stack)**

**Movie Review System: 4 - 5 Hours**

Create a Movie Review System – **Web Application**  
 **Tech Stack:**

* **Frontend *– Angular or any other UI framework***
* **Backend *– Java (Spring Boot) / Node.js***
* **DB *– MySQL (or any relational database)***

**Users/Roles:**  
Roles: User, Admin**(optional)**  
  
**Login:**  
1. User and admin should be able to login to the application through same portal  
2. Handle authentication using JWT **(Optional)**

**Assumptions:**No Signup required - assume demo users in database.  
Assume role for a user in the database.  
Upon successful login – user should land on the dashboard  
  
**User Module:**

* Once the user is logged in, display the list of all movies posted by the admin.
* Movies should be listed as cards with Image, Title, and overall Rating.
* User should be able to search/filter movies by title and genre.
* On Clicking a particular movie, it should display complete details about the movie like Title, Plot, Cast & Crew, Genre, Release Date, Language, **Collective Rating** etc. (Movie detailed view)
* Movie Rating
  + In the movie detailed view, there should be a comments/rating section where the user can type in his review and rate the movie out of 5 (Refer the below screenshot for how the rating should look like)
  + User should be able to like or dislike other user’s comments **(Optional)**
  + User can edit/delete the comments/rating he has given for a movie  
     A picture containing drawing

    Description automatically generated

**Admin Module (Optional)**

* Admin can view list of all Movies with the ability to search.
* Add a new Movie with required details like Title, Plot, Cast & Crew, Genre, Release Date, Language etc.
* Update / Delete existing Movie.
* Ability to delete comments posted by any user.

**Secure APIs/Pages:**

* A user should NOT be able to access Admin APIs.
* A user should NOT be able to view Admin Pages.

**Hosting: (Optional)**

Optionally you may host the application in AWS Cloud and share the URL. While hosting to cloud is optional, we expect proposed solution to be designed and developed as cloud native and eventually be able to deploy to cloud (preferably AWS). Be prepared to discuss alternate approaches with cost and/or scalability in mind.

**Evaluation Process**:

* Best practices in code
* User Experience/Attractive UI
* Completion of given use case
* Attempted optional tasks.

**Submission Guidelines:**

1. Share the public repository URL (GitHub or any other repo) with instructions to run in a Readme file. (if there are any secrets needed to run the application, email them beforehand)