

Coding Exercise - Full Stack Developer

Generic Instructions:

- Use any frontend or backend technology of your choice
- Should be accompanied by unit tests & preferably written using TDD approach
- Should commit code to a public git repository (GitHub) under a public handle
- Use generic package names; don't reference Maersk or any other Maersk brand
- Commits should be incremental so that one can look at the commit log and make sense of how the code has progressed along with the test cases. (recommended at least up to 20 commits to show how the code progresses; larger number of commits isn't a problem)
- Explain clearly on how to run both the backend and the frontend in the submission
- Don't spend more than 3 hours - 4 hours.

Purpose:

The purpose of this assignment is to assess the ability of developer to

- Develop frontend using a popular modern responsive UI framework (React, Vue, Angular, etc...)
- Develop backend with any technology of your choice
- Write test for both UI and backend code
- Follow clean code practices
- UX consideration

TASK:

- ❖ Create a web application to list movies available for a user.
- ❖ User should be able to search through the movies using the title and sort the result.
- ❖ When a selection is made it should go to the details page and present all the information available for that movie(Plot, Ratings, Poster, Stills etc). The user should be able to go back to the main page as well.
- ❖ There should also be a dedicated filter for the movies by Language or Location (drop down).

Bonus:

- Book a movie through the application
- Implement Login

Use the json file "**movies.json**" provided along with this instruction as a base model for your UI and API. You can add fields or append data on top as per your needs as long as the basic points listed in the **TASK** is covered.

FRONTEND:

You can use any front end libraries you are comfortable with. Follow standard folder and resource structures so that it is easy and simple to follow through the code.

Have a README.md file on how to build and run the UI application.

BACKEND:

Initially you can use one of the popular *fake server* to quickly ramp up the UI however you should **create your own API with the technology of your choice before the final submission.**

<https://github.com/typicode/json-server> (for a full fake REST API)

```
> npm install -g json-server
> json-server --watch movies.json
```

<https://github.com/marmelab/json-graphql-server> (for a full fake GraphQL API)

```
> npm install -g json-graphql-server
> json-graphql-server movies.js
```

Attached File (movies.json and movies.js):



movies.json



movies.js

Movie Sample:

```
{
  "Language": "ENGLISH",
  "Location": "DELHI",
  "Plot": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.",
  "Poster": "https://m.media-amazon.com/images/M/MV5BMDFKYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRlYWwMwFmXkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_QL50_.jpg",
  "SoundEffects": [
    "RX6",
    "SDDS"
  ],
  "Stills": [
    "https://m.media-amazon.com/images/M/MV5BNTYxOTYyMzE3NV5BM15BanBnXkFtZTcwOTMxNDY3Mw@@._V1_UY99_CR24,0,99,99_AL_.jpg",
    "https://m.media-amazon.com/images/M/MV5BNzAwOTk3MDg5MV5BM15BanBnXkFtZTcwNjQxNDY3Mw@@._V1_UY99_CR29,0,99,99_AL_.jpg",
    "https://m.media-amazon.com/images/M/MV5BMDFKYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRlYWwMwFmXkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_UX182_CR0,0,182,268_AL__QL50_.jpg"
  ],
  "Title": "The Shawshank Redemption",
  "imdbID": "tt0111161",
  "listingType": "NOW_SHOWING",
  "imdbRating": "9.2"
}
```