Novibet
Athens, Greece
www.novibet.gr
(123) 456-7890

Angular Interview Project

OVERVIEW

During this period of recruitment, we are seeking for enthusiastic and talented software engineers, who will participate in the design and development of enterprise scale applications as part of a skilled, supportive, and diverse team. The role of the software engineer is to build high-quality software that complies with coding standards and technical design. This project, as part of the recruitment procedure, will help us understand your skills, and figure out the exact position you will fit better.

GOALS

The purpose of this project will be the search, display and categorization of movies into user defined collections. The application will be based on a free API, which will supply us with necessary movie data. At first, the user will be able to search and get a list of movies along with their details. Then, they will be able to select a couple of them and store them into a custom collection. In addition, they should be able to view and edit any collections they have created.

This project is divided into three main sections of development.

1. **Search Page**
    a. This must be the home (default) page of the application
    b. This page must contain a simple input where the user will be able to write a keyword to search for movies. Furthermore, custom validation logic must take place into an angular directive with the following two validation rules:
        i. Minimum three (3) characters
        ii. Allow only alphanumerics
    c. Search results must be displayed using a pagination logic, and the following details must be shown:
        i. poster_path
        ii. title

iii. Vote_average

d. User must be able to navigate to MovieDetailsPage by clicking on the movie.

2. **Movie Details Page**
    a. This page must be accessible through angular routing.
    b. This page must be opened into a popup.
    c. This page must fetch and display the following movie details: *title, overview, poster_path, budget, release_date, revenue, vote_average, vote_count, spoken_languages*
    d. User must be able to post a rating for this movie. In order to accomplish this task, you need to get the sessionId for a guest user.

3. **Movies Collections Page**
    a. This page must contain all the available movies collections of the user (store them in browser's local storage), each collection contains a list of movies.
    b. User can navigate to collection create page. The needed information in order to create a collection is: *title and description*.
    c. By clicking to a collection a new page should open with the list of collection movies from where user can remove a movie from the collection or see the movie details using MovieDetailsPage mentioned above.
    d. User can add one or more movies to a collection by selecting the movies from the search results (SearchPage), then a popup should be opened with all the collections and user should select in which collection to add the selected movies.

SPECIFICATIONS

- The application should be based on the following free API:
  API reference: https://developers.themoviedb.org/3/getting-started/introduction
  API key: 85204a8cc33baf447559fb6d51b18313
- Use **Angular Material** or other UI kit (optional)
- Angular Version: Angular minimum version: 11.0.0 or latest.
- Nodejs minimum version: **14.0.0**