

Frontend Web Developer Assignment

We would like you to build a movie catalogue website based on the data provided by The Movie Database (TMDb). TMDb provides API for the largest catalogue of movies. The goal of this assignment is to build a website where users will be able to find a listing of movies based on different movie genres.

Sources of Data

In order to build the movie catalogue website, you need to use the following sources of data:

1. The API mentioned in the following section
2. Any other source of data, that you may find suitable

API

There is a public API available for retrieving information about a movie, for example:

https://api.themoviedb.org/3/movie/862?api_key=cd890f94a756b1518a2a17617a5b430e

https://api.themoviedb.org/3/movie/862/credits?api_key=cd890f94a756b1518a2a17617a5b430e

You can use the above API key. Or you can also create your own API key from here:

<http://www.desinerd.co.in/create-tmdb-api-key-add-elysium-kodi-addon/>

You can find this API documentation in detail here:

<https://www.themoviedb.org/documentation/api>

<https://developers.themoviedb.org/3>

Phase 1

We will start by building the front page of the movie catalogue site. The URL of this frontpage could

be something like this:

<http://localhost/movie>

or

<http://www.penta-movie-mania.net/movie>

The front page should have the following functionalities:

1. Listing of all the movies of the last 2 months(including the current month) on the TMDb API.

- The List item will be populated by scrolling.
- Users can change the initial Date Range to get the movie list between a certain period.

2. Random 5 movie names along with their poster image for each movie genre.

- A poster image of a movie can be retrieved using TMDb API

Phase 2

In this phase, we will need to implement the movie detail page. The URL of a movie detail page

could be as follows:

http://localhost/movies/{movie_id}

or

http://www.penta-movie-mania.net/movies/{movie_id}

The movie detail page should at least display the following information about the movie:

- Title
- Overview
- Poster image
- IMDb Link
- Rating
- Cast
- Crew
- Related movies (movie title, poster image)

Once you are done with implementing the movie detail page, make sure to link all the movies of the front page and movie genre page to the corresponding movie detail page.

Phase 3

In this phase, we would like to implement movie “watchlist” or “wishlist” functionality in our site so that, a user can add any movie to his wishlist/watchlist from the front page or movie detail page. We can have some “add to watchlist” icon or link (which might be similar to IMDb “+” icon) besides a movie poster or movie title. We would also like to implement a page, where a user can find a listing of all movies on his/her watchlist, sorted by adding date. The URL of this page could be as follows:

<http://localhost/watchlist/>

or

<http://www.penta-movie-mania.net/watchlist/>

You are free to use any suitable storage (DB, local storage, states, etc.) to implement this functionality.

User Experience

Your web application should be responsive, user friendly and visually pleasing. Also, make sure,

the application loads as fast as possible for a better user experience.

Expectation

From this exercise, we will try to understand the experience & working process of a candidate. We hope that candidate will properly use custom hooks, middleware and other good practices of ReactJS and TypeScript.

We also encourage the candidate to use popular ReactJS tools for state management like Redux or MobX.