Grace Joseph
Homework 8

8 / Implementing My Final Project

github repo
github pages

PART 1: Overview of “IamDb”

For my final project, I really wanted to create a site I have been wanting to create for a long time. As a child I would watch maybe 2 or 3 movies a week with my parents and always wanted to store my thoughts and opinions on them somewhere. My friends were always asking me for recommendations on films to watch and at one point they thought of me as their personal “IMDb” to which I would jest “yes, I am Db”. This project finally gave me an excuse to create an interactive tool / film blog where I can not only serve as an amateur film critic but also help users determine what films are worth spending time watching given their own interests and the information/insight I can provide them with. I want to not only convey my own opinions and thoughts on different films but also add easy access to important metadata related to each film such as the cast, writers, directors, genres, box office, content rating, etc. My target audience with creating this tool is anyone interested in finding new movies to watch based on their interests and/or interested in reading reviews and reflections on different aspects of films they may have already seen.

PART 2: Interacting with “IamDb”

- ➔ On the home page, users can interact with a carousel composed of layered film posters
  - ◆ Click on the left and right arrows in the middle section of the index page
- ➔ On the recommendations page, users can click on the different genres, platforms, and content ratings shown to filter the movies seen below

- Click on any of the gray squares to set a filter constraint and click on them again to undo the filter

➔ If you click on any of the movie posters on the recommendations page, it will redirect you to that film's details page

- Click on any of the movie posters (Even in the top 10 section) on the recommendations page

➔ If you click on any of the review rows on the reviews page, it will redirect you to that film's details page where my review will be below

- Click on any of the review rows in the reviews page

PART 3: Implementing "IamDb"

➔ **Movie Database API** on rapidAPI.com

➔ I chose to use this API because it made it really easy to create dynamic pages for each movie, such that I would only need to set an array of movie titles that I wanted on the site and then fetch the IMDb id for each film and fetch the metadata associated with it for each film using that id. The response to the API call was copious in information about the film and made it simple to template out a film details page.

➔ My code was structured so that I would first fetch the IMDb user ids for each film in my film array, using the name of the film as a parameter and always choosing the first result of the response and then using that id to search for the metadata for that particular film which I would then store in localStorage so that I wouldn't have to keep fetching the data as I reloaded the pages.

➔ Using this webAPI allows me to include a lot more films much faster on the site than if I were to manually input the data for each one and worry about making the pages dynamic. I didn't have to find individual posters or separate synopsis

for each film either since part of the response included a brief plot as well as some images of the film.

PART 4: Iterating on “IamDb”

When implementing the ultimate site, I iterated by adding the additional filter of content rating in the recommendations page as well as adjusting the data shown in the film details page. I also changed the reviews page to only showcase the films currently with reviews.

PART 5: Challenges with “IamDb”

When I chose this project, I knew that it would be difficult to have to manually input all the reviews for each film. Because I obviously didn’t have time to write 54 reviews, 1 for each of the films I chose to include in the site currently, I knew that I would have to change the view of the film details depending on whether I had written a review yet. In addition to that I had to balance having the data for each film completely automatic through the API while also have the more subjective data such as my ratings, the bechdel test score, and the availability of each film across streaming platforms (i tried different APIs for this but to no avail without a significant price tag) fully manually inputted into the system in javascript through classes.