REST API IMPLEMENTATION

This document goes over the process of implementing a Rest API with the use of modules and local storage.

1) Index.html

This file simply defines the layout of the webpage. The html file provides a framework for designing a single page webpage with the use of JavaScript. The file mostly consists of "ids" and "classes" that can be used by CSS stylesheet and JavaScript to modify the contents of the webpage. This file also imports a module which prevents multiple imports of JavaScript files and helps for a better design practice.

2) Styles.css

This file is simply a stylesheet for the frontend of the webpage. It was used to implement tiles for each movie, the hover effect for the overview and the ratings. One of the important features of the app is to be able to search for a movie and fetch the data from the API. Thus, a search button was designed using the CSS features.

3) Scripts

a) http.js

showMovies(data)
This function simply fetches the title, poster, rating and the overview of a movie from the API and sets it to a constant called movie. We then edit the innerHTML to display the data fetched from the API.

getMovies(url)
This function takes in the API URL and fetches the data from the API. Once we have the data, we call the showMovies function which filters out the data we want and change the innerHTML for out frontend design.

b) recentButton.js

This file only contains a function expression that creates a new button when we search for a movie. As we click search, we want to make sure that the search term is stored in a variable and presented as a button. This is because we want to keep track of the recent searches which will appear as buttons that the user can click to get to their recent searches.

c) <u>view.js</u>

displaySearch(movieList)

While we want to implement buttons that help users to click on their recent searches, adding on buttons will make the webpage full of clickable buttons. Thus, we want to only display 5 of our recent searches. This method makes sure that we are only displaying the last 5 elements of the array that is used to store the search results. This way, we can prevent the cluttering of the webpage with a lot of recently searched buttons.

getcolor(vote)

This function simply changes the color of the rating. A rating of more than eight is colored green, between five and eight is colored orange and less than five is colored red.

d) <u>app.js</u>

This file contains the details of the API such as the API key, the base URL, search URL, etc. We also declare an array called movieList that will be used to store search data into local storage. We have an event listener for the search function of the application, and it pushes the data to the array and local storage as it is triggered. Finally, since we are using modules, we have to make sure we trigger window.start to our script is triggered when the browser window loads.