Discount Blockbuster

Connor, Elle, Jim

Our Idea

Discount Blockbuster - your one stop shop for looking up your favorite movies!

- Search for all movie titles on IMDB
- Get information about all your favorite movies
- Find out what new shows and movies are streaming

Code Presentation

Code Review - Promise All. Map

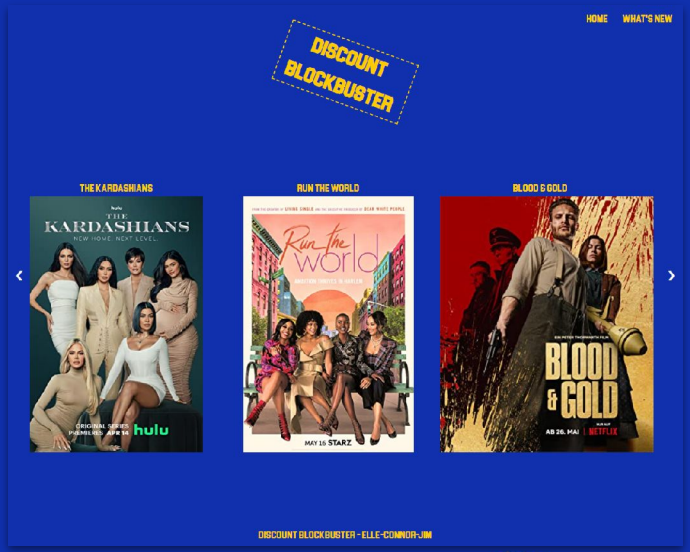
```
searchBarSubmit.addEventListener("click", function(event)  {
    event.preventDefault();

    const searchBar = document.querySelector("#query").value;

    const URL = `https://api.watchmode.com/v1/search/?apiKey=bkzyP0dAymwzKoyjlBNVHHCld7cFfWXlwegOwfHr&search_field=name&search_value=${searchBar}`;

    const response = fetch(URL).catch(err => console.log(err));
    let searchResults;

    response
        .then(response => {
            if(response.status !== 200){
                throw Error('Media not found')
            }
            while(resultsList.firstChild){
                resultsList.removeChild(resultsList.firstChild);
            }
            while(document.querySelector('.errorMsg')){
                document.querySelector('.errorMsg').remove();
            }
            return response.json();
        })
        .then(movieData => {
            searchResults = movieData.title_results;
            if (searchResults.length === 0 ) {
                throw Error('Media not found')
            }
            return Promise.all (searchResults.map(r => getPoster(r.imdb_id)))
        })
        .then(posters => {
            for(let p = 0; p < posters.length; p++){
                searchResults[p].poster = posters[p];
                populateItem(searchResults[p]);
            }
        })
        .catch(function(error) {
            populateErrorItem(error);
            console.log(error)
        });
});
```

Promise.all takes in the results of our Movie Data object from the watchmode API. Each objectin the Movie Data array is passed to the getPoster function. The getPoster function uses the imdb_id key to call the OMDB API to retrieve the poster URL.

The return from posters is then passed to populateItem function which places the posters on the page

Code Review - Carousel Code

```
const forwardBtn = document.querySelector('.forward');
const backBtn = document.querySelector('.back');

forwardBtn.addEventListener("click", function(event) {
    showSlides(slideIndex);
})

backBtn.addEventListener("click", function(event) {
    plusSlides(-6)
})

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function showSlides(n){
    let slides = document.getElementsByClassName("whatsNewMedia");

    for (let i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }

    if (n >= slides.length) {n = 0}
    if (n < 0) {n = slides.length -3}

    for(let j = 0; j < 3; j++){
        slides[n].style.display = "block";
        n++
    }
    slideIndex = n;
}
```



DISCOUNT BLOCKBUSTER

Challenges We Faced

- Making nested fetch calls for multiple APIs
- Local Storage
- Carousel functionality
- CSS formatting
- Finding free working APIs to use

What We Learned

- Promises, promises, promises…
- Using different APIs
- Passing data between HTML pages
- Commit, Push, Merge, repeat

What We would've done differently

- Utilize Bootstrap as a styling foundation
- Use axios in place of native fetch calls
- Research third party libraries for a carousel feature
- Find a better API to use for our data

What Else We would Add

Future Features on our backlog

- Additional functionality to search bar
- Add a details page for people(actors, directors, etc)
- Create a contact us page
- Separate currently streaming by streamer service
- Sort streaming by movies and tv shows
- Add movie trailers to movie details