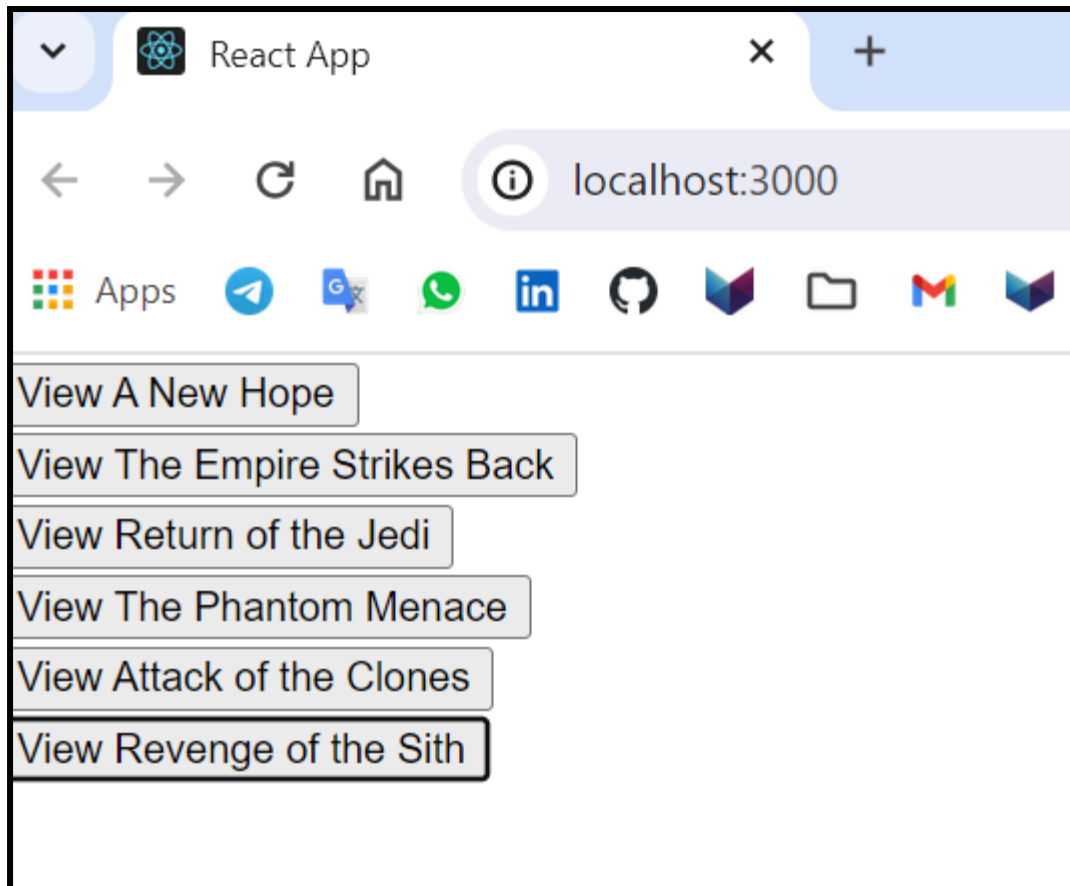


Star Wars Favorites

Tasks

1. ****Debugging:****

- Start the application and identify the 2 errors introduced.



1. Buttons are clicked but nothing happens, often referred to as a "non-responsive button" or "unresponsive button."
2. Buttons appear on the very left side of the page, are almost cut off, and are greyed out in a React application running on

localhost:3000 can be caused by several factors related to CSS styling, layout, and HTML structure.

- Dive into the codebase and pinpoint the source of each error.

1.

```
<MovieDetails movie={selectedMovie}
favorites={favorites} />
```

It seems that the `onFavoriteToggle` function is not being passed correctly to the `MovieDetails` component. This absence would indeed cause the buttons within the `MovieDetails` component to be unresponsive, as they rely on this function to toggle the favorite status.

2.



```
index.css x
src > index.css > body
1 body {
2   margin: 0;
```

The `margin: 0;` rule in the `index.css` file sets the margin of the `body` element to `0`, which removes any default margin applied by the browser. While this rule can be beneficial for achieving consistent spacing and layout across different browsers and devices, it may also affect the layout of elements within the body, including buttons.

- Rectify each error ensuring the application displays and functions correctly.

Error 1:

Non-responsive Buttons

Issue:

The `onFavoriteToggle` function is not being passed correctly to the `MovieDetails` component.

Solution:

Passing the `onFavoriteToggle` function from the parent component (`App.js`) to the `MovieDetails` component as a prop.

1. **Verifying Function Definition and Passing in `App.js`:** Ensuring that the `handleFavorite` function is defined correctly in `App.js` and is passed as a prop to the `MovieDetails` component.

```
function handleMovieSelect(movie) {  
  setSelectedMovie(movie);  
}
```

changed to

```
const handleMovieSelect = (movie) => {  
  console.log("Movie selected:", movie);  
  setSelectedMovie(movie);  
};
```

`onFavoriteToggle={handleFavorite}` **was added in the return scope**

2. **Ensuring Correct Prop Usage in `MovieDetails.js`:** Verifying that the `onFavoriteToggle` prop is correctly used in `MovieDetails.js`.

NO CHANGES

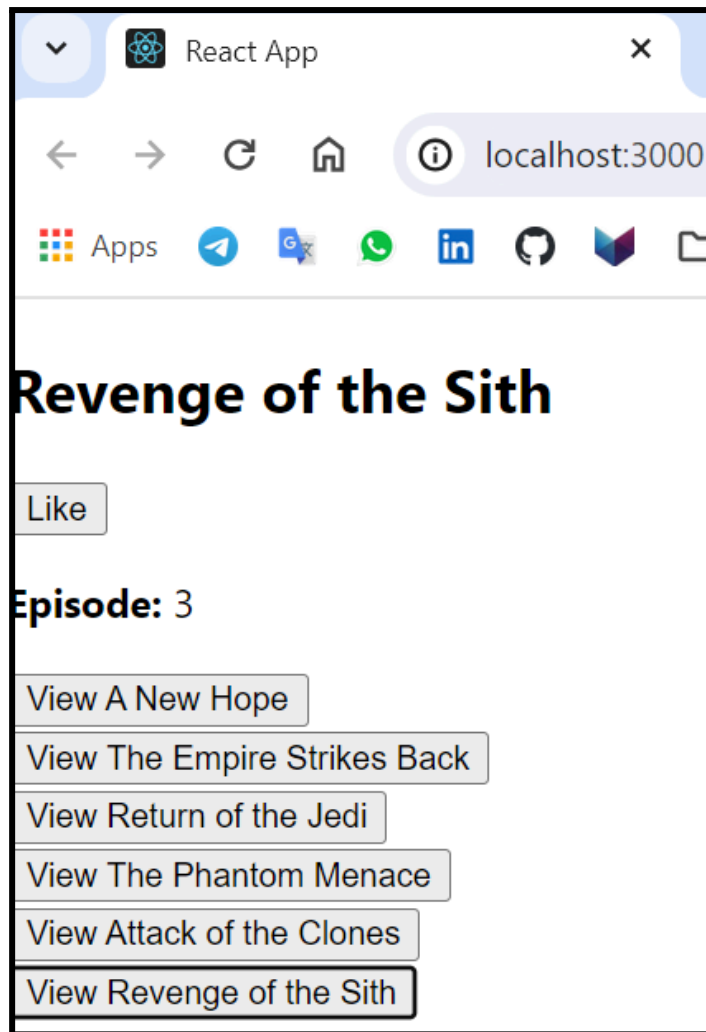
3. **Checking `MovieList` Component for Correct Movie Selection:**

Ensuring that `handleMovieSelect` in `MovieList` is properly implemented.

`onMovieSelect={onMovieSelect}` **changed to** `onMovieSelect={() => onMovieSelect(movie)}`

4. **Checking `MovieItem.js`:** Making sure that the `onMovieSelect` function is passed the correct movie object.

`<button onClick={() => onMovieSelect()}>` **changed to** `<button onClick={onMovieSelect}>`



Error 2:

Buttons appear on the very left side of the page, almost cut off, and greyed out.

Issue:

The `margin: 0;` rule in the `index.css` file sets the margin of the body element to 0, which removes any default margin applied by the browser.

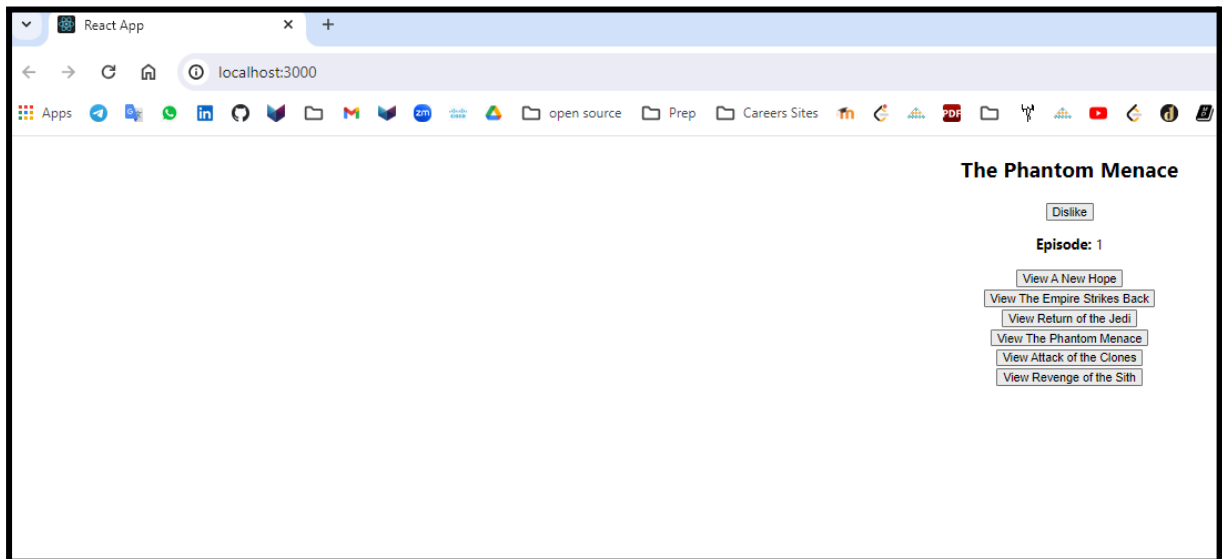
Solution:

Adding CSS styles to ensure that the buttons and their container elements are properly positioned and styled. Specifically, ensure that the container elements have appropriate padding or margin.

Importing App.css in App.js:

This ensures that the styles defined in App.css are applied to the components rendered within App.js.

`import './App.css';` // Importing the CSS file **was added in App.js file.**



- Offer a concise explanation of the root causes and detail your solutions.

Error 1: Non-responsive Buttons

- Root Cause:

The `onFavoriteToggle` function was not being passed to the `MovieDetails` component in `App.js`. Without this function, the buttons within `MovieDetails` had no functionality, leading to non-responsive behavior when clicked.

- Solution:

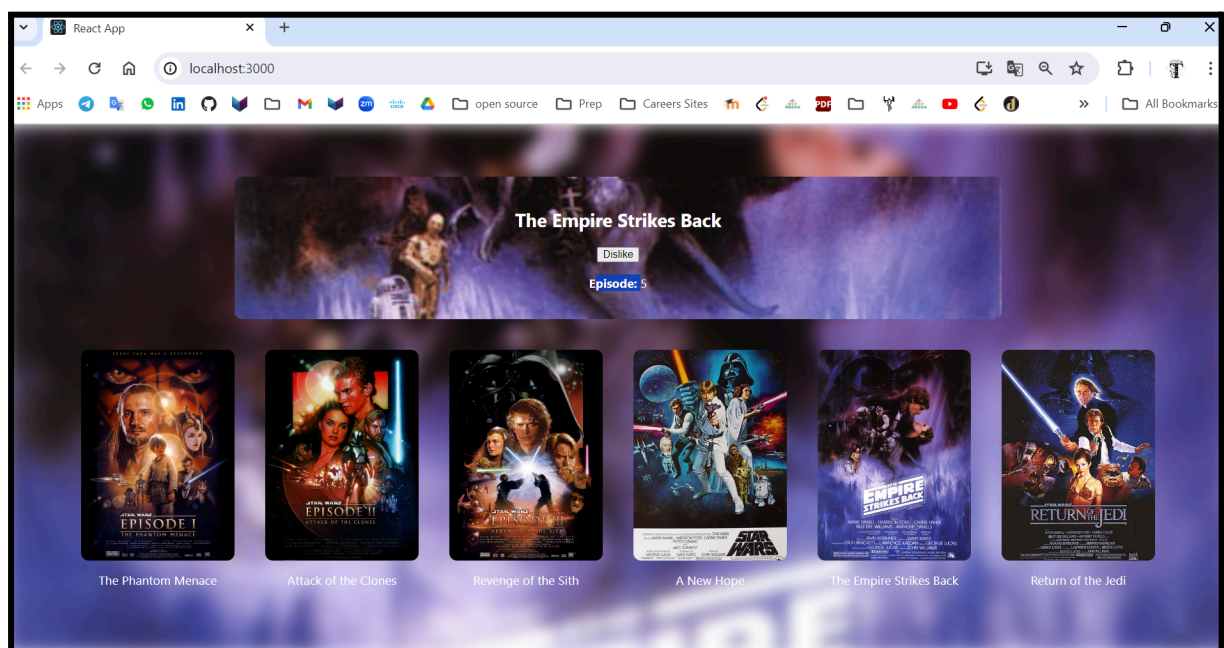
Passing the `onFavoriteToggle` function as a prop to `MovieDetails` in `App.js` as described above.

Error 2: Buttons appear on the very left side of the page, almost cut off, and greyed out

- Root Cause:
The App.css file, which contains the necessary styling for button alignment and appearance, was not imported into App.js. As a result, the default browser styling was applied, causing layout issues.
- Solution:
Importing App.css into App.js to ensure that the styles are applied to the buttons and other elements.

2. ****Styling:****

- Refactor the application's styling to match the provided [mockup](page_mockup.png)



This is the current state of the application. I am aware that further modifications are necessary to achieve an exact match with the provided

reference image. However, these are the adjustments I could implement within the allotted time.

Additionally, please note that the images for the movies "The Empire Strikes Back" and "The Phantom Menace" are different from those in the reference image, as they have been updated on the website.

3. ****Improvements:**** (Optional)

- After implementation of the basic design, you can go ahead and implement any other enhancements or features you feel would optimize the application.

To enhance user experience, I would propose the following enhancements and features to optimize the application:

1. **Sorting and Filtering Options:** Implementing functionality to allow users to sort and filter the movie list based on criteria such as genre, release date, rating, and user preferences. This empowers users to customize their viewing experience according to their preferences.
2. **Search Functionality:** Integrating a search bar to enable users to quickly find specific movies by title, actor, director, or keywords. This enhances navigation and facilitates the discovery of new content.
3. **User Profiles and Preferences:** Introducing user profiles where users can personalize their experience by setting preferences, such as preferred genres, favorite actors, and languages. The application can then tailor movie recommendations and content suggestions based on these preferences.
4. **Rating and Review System:** Allowing users to rate and review movies they have watched, providing valuable feedback to other users and fostering community engagement. Incorporating a rating system can also influence personalized recommendations and improve content discovery.

5. **Watchlist and History:** Enabling users to create a watchlist of movies they intend to watch in the future and maintain a history of previously viewed movies. This feature facilitates organization and tracking of viewing habits, enhancing the overall user experience.

6. **Social Sharing and Recommendations:** Integrating social sharing functionality to allow users to share their favorite movies with friends on social media platforms. Additionally, leveraging social connections to provide personalized movie recommendations based on friends' viewing history and preferences.

7. **Responsive Design:** Ensuring the application is optimized for various devices and screen sizes, including desktops, tablets, and smartphones. Implement responsive design principles to deliver a consistent and user-friendly experience across all platforms.

8. **Accessibility Features:** Incorporating accessibility features such as keyboard navigation, screen reader compatibility, and color contrast adjustments to ensure the application is accessible to users with disabilities and diverse needs.

9. **Clearer Like and Dislike Button:** Enhancing the Like and Dislike button by replacing the toggle with a visually intuitive design. Utilizing a white heart icon with a black outline, transitioning to red upon selection, similar to popular social network interactions. This modification provides users with a clearer indication of their preference for each movie.

10. **Visual Indicators for Favorite Movies:** Introducing visual indicators for movies marked as favorites to provide users with quick access to essential information without the need for interaction. Implementing a subtle frame surrounding the movie poster, accompanied by a small heart icon positioned adjacent to the frame. By integrating these visual cues directly onto the movie thumbnails, users can effortlessly discern favorite movies, enhancing overall accessibility and user-friendliness.

These enhancements collectively aim to optimize the application's usability, engagement, and accessibility, providing users with a seamless and enjoyable movie browsing experience.

4. ****Upload to github:****

- Once finished, upload the project to github.

<https://github.com/Tamara1000/StarWarsFavorites>

Thank you for the opportunity to complete the task—I thoroughly enjoyed it :)