Project 2 Documentation

Process

Tutorials Used:

Image Overlay (For Image Information Hovering):
[https://www.w3schools.com/howto/howto_css_image_overlay.asp]

WebStorage API (For Storing Search Terms):
[https://github.com/tonethar/IGME-235-Shared/blob/master/tutorial/web-apps-9.md]

GIF Finder (For Ways to Search APIs):
[https://github.com/tonethar/IGME-235-Shared/blob/master/tutorial/HW-gif-finder-lab.md]

Web Apps Lab 5 (For Radio Box Reference):
[https://github.com/tonethar/IGME-235-Shared/blob/master/tutorial/web-apps-lab-5.md]

Overall Process:

1. Start with base GIF Finder Homework.
2. Replace loading with Rick And Morty API
3. Add the filters
4. Add next and previous buttons.
5. Add WebStorage API requirement
6. Apply CSS
7. Apply image overlay CSS
8. Clean up code

Citations

Images

Rick & Morty Banner: [https://justtoysintl.com/collections/rick-and-morty]
Portal Background: [https://wallpaperboat.com/rick-and-morty-portal-wallpapers]
Portal GIF: [https://giphy.com/stickers/adultswim-emoji-i2tLw5ZyikSFdkeGHT]

API / Character Information

[https://rickandmortyapi.com/]

Features to Notice

- Hovering over images provides information about the character.

How I Met the Requirements

1. Functional
    a. I selected a specified API option.
    b. I saved the last search term in local storage.
    c. There are three controls:
        i. Search Field
        ii. Filter Section
        iii. Previous/Next Buttons
    d. No JavaScript errors
2. Design & Interaction
    a. The interface doesn't closely resemble the GIF Finder.
    b. Widgets are correctly labeled.
    c. The app is intuitive.
    d. When searching with an empty box, the results section reports that the user needs to type something in the search field.
    e. The user knows the app's state through the status section in results either through the loading portal or a written status.
    f. The app is mostly responsive, although it isn't perfect.
    g. Flexbox and grid help to organize the page.
3. HTML/CSS & Media
    a. HTML Validated
    b. CSS Validated
    c. CSS is external
    d. Semantic structures used
    e. Images are optimized
4. Code Conventions
    a. External JS
    b. let and const used
    c. querySelector() and querySelectorAll() used
    d. D.R.Y. principles followed
    e. Variable names and functions start with a lowercase letter
    f. Code is well-commented
    g. console.log() commands commented out
5. Milestones
    a. Proposal submitted on time
    b. The prototype presented on time, although missing the prev/next function
    c. Final submission submitted with full functionality
6. Documentation
    a. Well-documented

The Grade I Believe I Should Receive

I believe I should receive an A. I think I received the following points in each section:

- 10 points in the API section of the rubric
- 10 points in local storage
- 10 points in controls
- 10 points in the error-free portion
- 10 points in code
- 4 points in the semantic structure
- 4 points in HTML validity
- 4 points in CSS validity
- 7 points in visual design because I believe the top section should be better presented, although my crit group didn't give me any feedback on that portion, and I wasn't sure how to improve it
- 8 in interactive design
- 7 in responsive design since the header image could use improvement
- 6 in checkpoint deliverables as the prototype was missing the third function
- 6 in documentation

These values total to 96, which is above the 90 point threshold for an A.