Part 1

The purpose of my website is to show the final results/standings of the 2022 F1 Season for both the Drivers and the Constructors. The Driver's Standings and Constructor's Standings pages both show the final results as well as some additional information in the dropdown menu. The Fun Visualizations page shows the cumulative points total of each driver as the season progresses. It is interesting and engaging for newcomers of F1 to learn a little bit about how the 2022 season went and to learn a little bit about each driver and their teams. In terms of resizability/scalability, both the Driver's and Constructor's standing pages should work at both 1200px and 460px widths.

Part 2

NOTE: 1. Ergast API access seems to be blocked when opening it through GitHub (was loaded over HTTPS, but requested an insecure XMLHttpRequest endpoint) but not locally. I've gotten around this by In both clicking the lock icon next to the URL \rightarrow "site settings" \rightarrow "insecure content" \rightarrow "allow". 2. Ergast API pull requests are sometimes blocked for me. The Constructor's standings page is coded manually and is basically the same as Driver's Standings which uses data from the API.

- On the Landing Page, users can click through the image carousel or let it play since it is on a timer.
- Driver's Standings and Constructor's standings, the user can click on the horizontal tabs to reveal more information.
- The user can also click the hamburger menu in the upper left to reveal other pages. To exit the hamburger menu, the user can click on the hamburger menu or the "x".
- On all pages, the user can click on the F1 Logo at the top of the page to return to the main page(Landing Page).
- On web, the cursor should change to a mouse upon hovering over the tabs, the hamburger menu as well as the F1 Logo to suggest interaction.
- For the Fun Visualizations page, I could not get the visualization to update its size dynamically. However, after reloading the page, the size of the visualization should update accordingly.

Part 3

Ergast.com API

- I needed information on the 2022 F1 Driver's points, wins etc. Ergast is a relatively comprehensive source of Formula 1 data which had all of this information.

- Ergast categorizes it's data so I sent an API request for the data I needed. I created a HTML template of the data I wanted to display and filled it with the Ergast data by using a constructor.
- This API adds the detail that a user needs to understand the Driver and Constructor standings for the 2022 F1 season.

D3

- I wanted a way to visualize each driver's points tally throughout the entire 2022 season.
 D3 was a good way of taking the points data and representing it visually.
- I initially wanted to use data from Ergast and have D3 read it, however, I encountered too many issues with that. Instead, I manually created a CSV that logged the cumulative points total of each driver after each race. Then I created a multi-line graph using D3.
- D3 helps the user to see the points disparity between the drivers and just how dominant of a season Verstappen had. It is also another way of understanding the driver's points standings information from the homepage.

Part 4

In the first few sketches, I was focused on ideas that I would actually want to develop or ideas that I would have some fun developing. So, I was pretty pleased that much of the feedback I got seemed to favor the F1 idea.

From there, the higher fidelity prototypes as well as the initial rounds of user testing focused on features that I would want to include. Core features like Driver's and Constructor's standings have not changed from this stage. I wanted to include a visualization early on as well, however, I needed to change the type of visualization because the data that I initially wanted to use was not readily available.

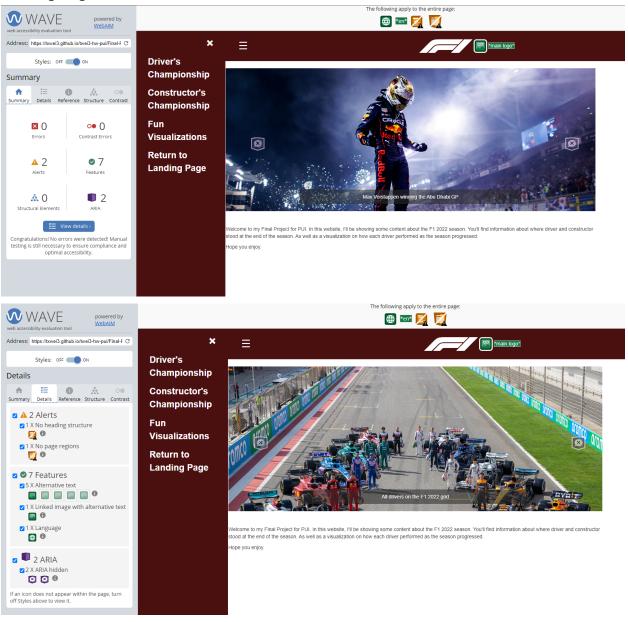
From here, most of the coding was about refining the visual design of the page; the feature set was largely set in stone. I did, however, add a landing page as a means of introducing people to the website and generally explaining the content they will see.

Part 5

It was my first time using an API as well as D3 so, I generally encountered a lot of challenges implementing what I wanted to do. Ergast had a tendency to not work - often blocking me from getting pull requests which was really frustrating. With D3, I would find examples online or follow video tutorials, however, I think the syntax wasn't compatible with the current version of D3 so large swaths of code were unusable.

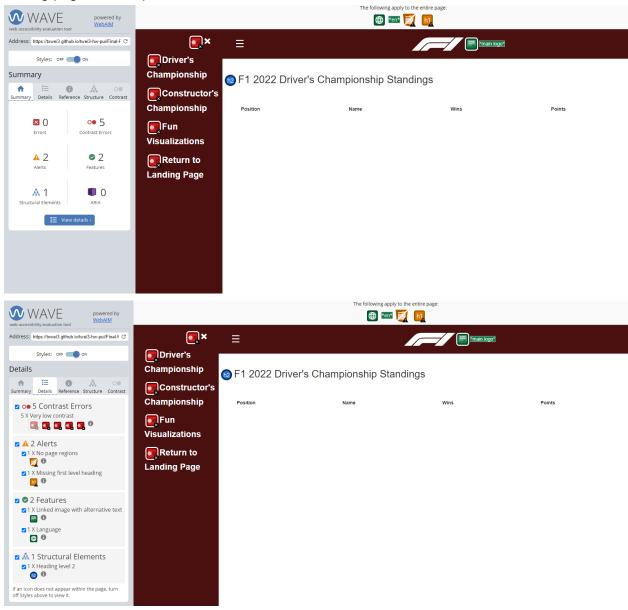
Screenshots of each page

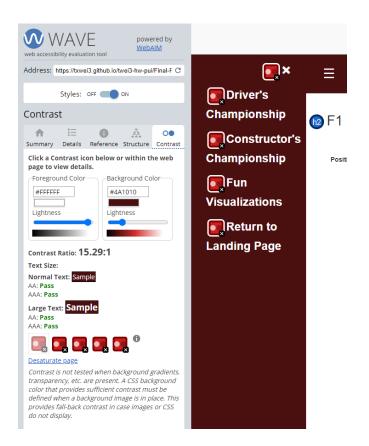
Landing Page



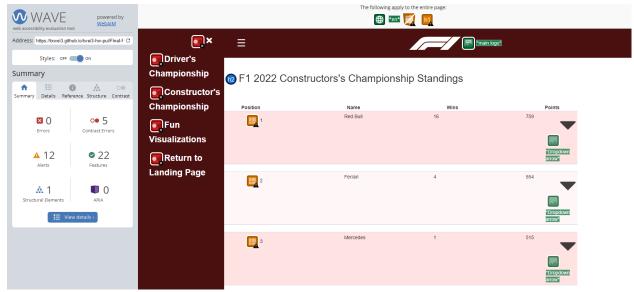
Driver's Championship

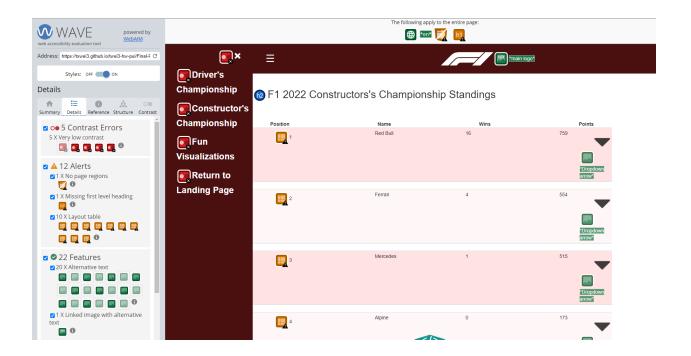
*I don't understand why there are contrast errors when the code is virtually identical to the landing page. It even passes the contrast test on WAVE as well...





Constructor's Championship





Fun Visualizations

