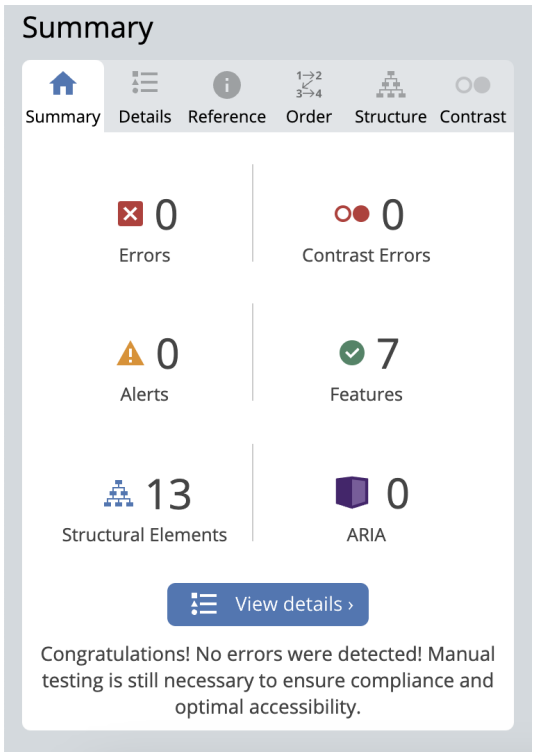
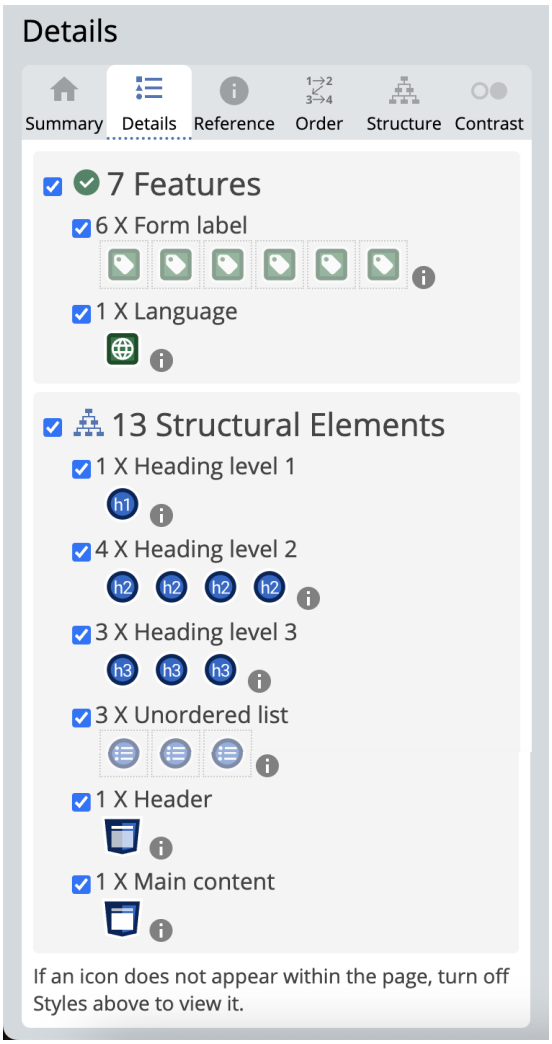
Screen size: The website was designed for desktop pages (any size), but also plays well on mobile devices. I have only tested it on iPhone 15 Pro.

WebAIM WAVE check:





Part 1:

My webapp is designed to streamline the process of tracking and planning workouts for powerlifting enthusiasts. It serves as a digital workout log, allowing users to record their exercises, sets, reps, weights, and exertion levels; the unique feature lies in its ability to rate the exertion of each set. This data plays a crucial role in providing progressive overload suggestions tailored for the user's next workout session.

The webapp boasts a minimalistic yet highly intuitive interface, supporting core powerlifting exercises like deadlifts, squats, and bench presses. A rest timer integrated into the top menu aids users in managing rest periods effectively between sets and exercises. The timer counts to 60s to allow for a sufficient break between exercises.

Various interactive elements keep users engaged with the webapp. The highlight is the dynamic rest timer, showcased through a visually appealing circular progress bar. Additionally, the inclusion of clickable exertion dots and responsive buttons enhances the interaction, with visual cues like shifting and dimming effects signaling their functionality. These features, coupled with the flexibility to add or delete workout sets, render the webapp not just a tracking tool, but a personalized fitness companion for each user.

Targeted at powerlifting enthusiasts who value structured and informed workout sessions, this webapp is a perfect blend of utility, interactivity, and personalization, reshaping the traditional pen and paper approach to workout tracking.

Part 2:

User Interaction:
Adding a Workout Set: Click the "Add Set" button in the exercise section to log a new set. Enter weight, reps, and exertion level, then submit.

Using the Rest Timer: Click "Start" on the rest timer to begin countdown, "Pause" to pause the timer, and "Reset" to reset it.

Navigating Between Exercises: Click on an exercise title to view and add sets for that specific exercise.

Deleting a Set: Click the "Delete" button under a set to remove it from the workout log.

Part 3:

External Tool: ProgressBar.js (https://kimmobrunfeldt.github.io/progressbar.js/)
Reason for Choice: I chose ProgressBar.js for its simplicity and visual appeal. It provides a clear visual representation of time, which is crucial for the rest timer in the workout app.
Usage: I used it to create a circular progress bar that acts as a rest timer, updating in real-time as the user rests between sets.
Contribution to Website: It enhances the user experience by providing a visually engaging and interactive way to track rest time, adding a professional touch to the site's design.

Part 4:

My initial prototype was simple: it used white tiles against a gray background. Each exercise tile had its own “add-set” input box, which was busy and redundant. I learned how to hide this input box until the exercise tile is selected, at which point it appears.

My second iteration of the prototype switched to a “dark mode,” which was the product of researching Google’s Material Design. I used a variety of dark shades to convey depth on the website, while the clickable buttons are vibrant and high-contrast. Initially, the buttons used a blue-green and purple variety, but I changed the colors to an orange and teal-blue as I felt they were more “active,” fitting in with the exercise theme.

After running it through WebAIM, I found several accessibility issues. I increased the contrast of several buttons, as well as organizing the headings to go in order (H1-> H2 -> H3).

Part 5:

When I first started experimenting with my site, I thought React would provide a great framework to dynamically update elements of this website as a single page application. After writing several components, I realized I was in over my head with React and opted to use vanilla Javascript, focusing on elements I learned during the semester. I struggled with implementing the progressbar.js since the documentation was initially difficult to find, but after some trial and error, I got it to do what I wanted.