Clara Cook
05-630
Assignment 8

Final Project

Part 1:

Music is a powerful force that we should all be able to enjoy. That being said, much of music relies on understanding the auditory differences between notes. For this reason, I built a tool that is geared towards two audiences: those who are hearing impaired or those learning about the relationships between sounds and pitches. The purpose of this tool is to allow users to record sound and see how that would be visually represented: it takes in user input of people recording small sets of sound and outputs a visual representation of the volume, timing, and pitches that were just taken in. It is interesting because it takes inspiration from people with synesthesia, those who associate multiple stimuli together, such as color and pitch. In this visualization, the website matches up each measurement with a volume and pitch to a y-coordinate and color.

Part 2:

- Interaction Implemented: Recording
    - Click the "Record" button on the left hand side of the homescreen.
    - It may ask you to allow recording.
    - Once this is allowed, notice that the recording should start (indicated by the "Recording" note with a blinking red light and counter), feel free to make noise while the recording is in progress.
    - Then click "Stop", which is in the same place as the previously mentioned recording button.
- Interaction Implemented: Pause/Resume
    - Hit the "Record" button.
    - Hit pause icon and wait for a second to see that the counter has actually paused
    - Hit the play icon and see that the counter resumes
    - Again, feel free to make noise while the recording is in progress
- Interaction Implemented: Play/pause your recording
    - Go to the right portion of the screen where little record bars have popped up
    - Click the play icon on the grey, rounded rectangle to hear recording
    - Click again to pause the recording
- Interaction Implemented: Download recording
    - Go to the right portion of the screen where little record bars have popped up
    - Click on the 3 dots at the end of the grey, rounded rectangle where you played your recording
    - Click download
- Interaction Implemented: scroll through graphs

    - Navigate to the graph section of the home page
    - Scrolls vertically to see both graphs you have made so far
- Interaction Implemented: Change which graphs are shown
    - Click on and off the checkboxes next to the list of recordings
    - Click both off to see both disappear
    - Click one off to see one disappear
- Interaction Implemented: Hover on graphs
    - Make sure at least one of your recordings is displayed by making sure one of the records has a blue checkmark next to it
    - If the recording had substantial sound, there should be dots on the graph
        - If there are no dots on the graph, try recording again (it does not work that well with headphone input and requires more sound)
    - Once there are dots, hover over a dot to see the note, volume, and time (rounded to the nearest second) that the measurement was taken
- Interaction Implemented: Plotly.JS controls
    - click to a section of the graph without dots
    - Drag the click to cover some dots to zoom in on that section

Part 3:
Describe what external tool you used (JavaScript library, Web API, animations, or other). Following the bulleted list format below, reply to each of the prompts. (I will stop reading at the 4th sentence, so please be concise)

- D3
    - I chose to use it because it wanted to create an interactive graph where users could gain more information by hovering over or doing something with the graph.
    - I created hover panels for each measurement on the graph to give further information about the note, volume, and time to the nearest second for each measurement.
    - It adds further interactivity to the graph and means that users do not have to read the axes of the graph to understand what each data point means.
- Plotly.JS
    - I chose it because it would allow me to render a graph in javascript, which was my overall language of choice. It also means that I could incorporate more interactivity than something like matplotlib, and I wanted to use something I had not used before.
    - I used it by creating my scatter plots with the data that I acquired. I also used it by placing in all of the controls, such as zoom and pan.
    - It adds the main component of my application. It is the basis for what I use to visualize the music and gives users the ability to focus on whatever part of the data they see fit.

- Web Audio API (in the bigger library of webkitURL)
    - I chose this because I did not know how to work in an auditory context on the browser. For this reason, when I was researching how to deal with notes on the browser, one of the main ideas was the “Audiocontext”, which this API uses quite effectively.
    - I used it by embedding the audiocontext in my recordings and took the byte strings produced in the audiocontext to figure out the pitches and volumes associated with each byte buffer. I also was able to use it to enable pause and play in my recordings because I could figure out where the recording stopped and started.
    - It adds the data to the application. Without this API, I would not have been able to get any of the data that I needed for my graphs, meaning that other than the time, I would not have had anything to visualize at all.
- CDN Recording.JS
    - I chose this library because it provides the necessary features to temporarily store audio in the browser and allows measurements to be taken in the middle of the different functions. Many recording APIs/libraries have functions that allow the user to record and stop recording, but this one allowed me to embed a lot of other functionality inside.
    - I used it by saving the audio a user records to the browser temporarily. This library is used to create the downloadable audio file.
    - It adds to my application because it provides a space for the web analysing to happen and it gives users a chance to relisten to what they recorded to match it up with the created graph. It also adds the ability to download the recorded file.
- Other libraries/APIS used only for styles:
    - Bootstrap
    - Fontawesome

Part 4:
I iterated on my HW7 mockups in a few key areas. I first found that condensing the graphs horizontally made them very difficult to read when there were more than a second or two of recordings, meaning that I changed that section to scroll, rather than lining them all up side by side. I also got rid of the decade by decade section because acquiring the rights to play songs is very expensive and billboard does not categorize the top songs of the decade, only the top songs of a given week.

Part 5:
The most difficult part was figuring out how to translate buffers of bytes into actual music notes. The set of steps was long and arduous and meant that I had to learn a lot about how frequency is

calculated from sin waves. It was also difficult to work with permissions for the audio input on different browsers.

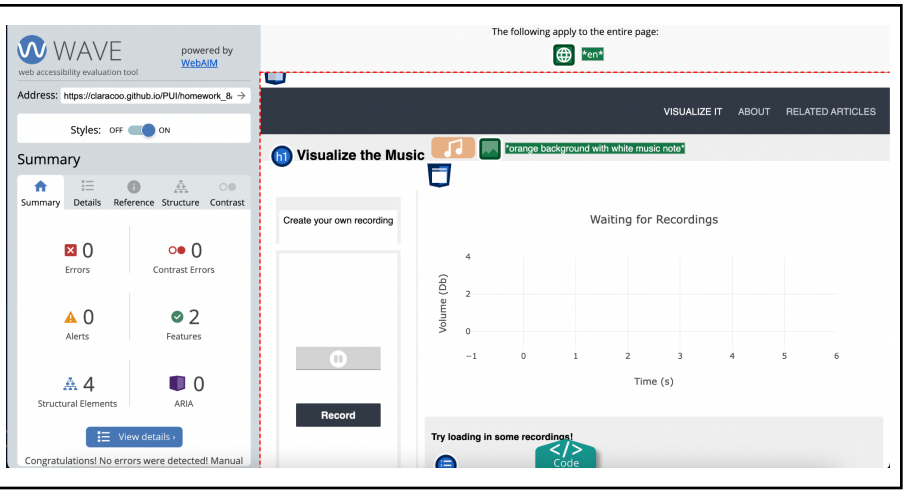
Additional Asks from the project document:
Github Link:
https://github.com/claracoo/PUI/tree/main/homework_8

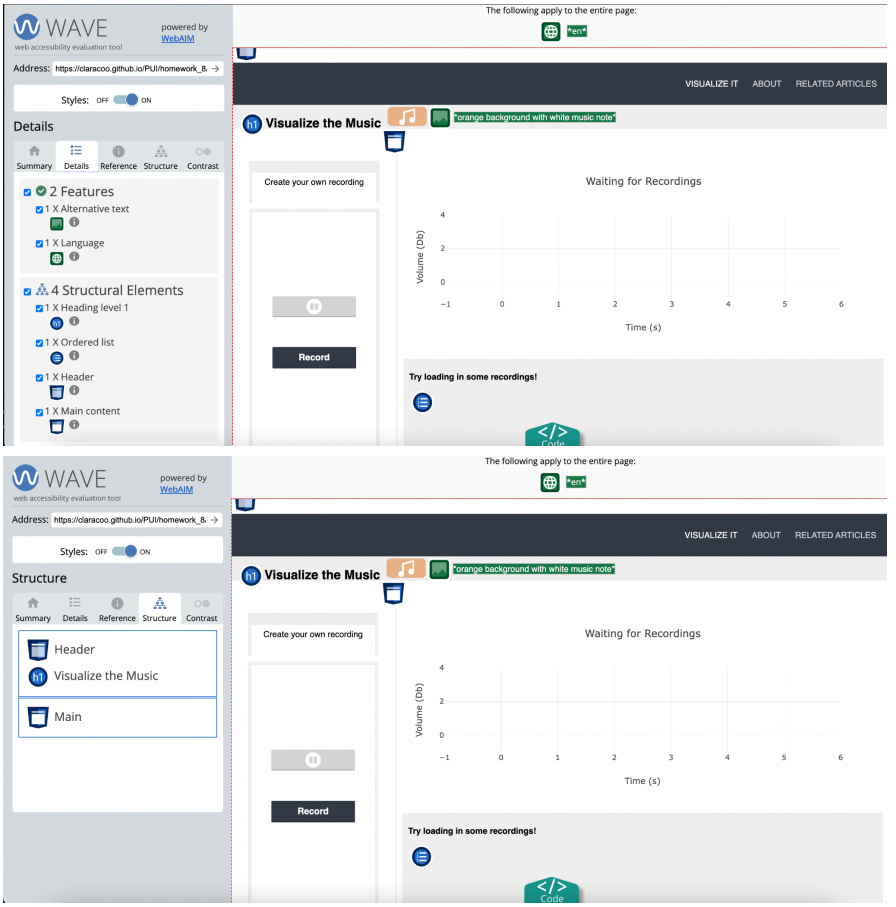
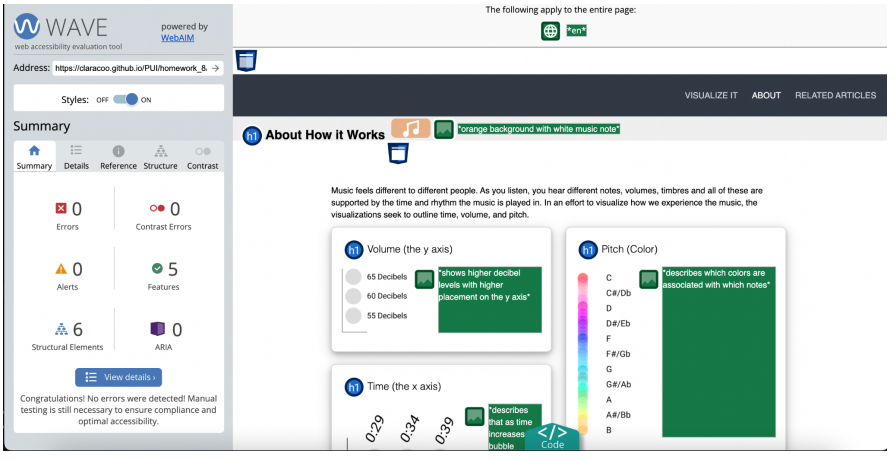
Netlify Deployed Link:
https://hopeful-khorana-b91448.netlify.app/homework_8/html/home

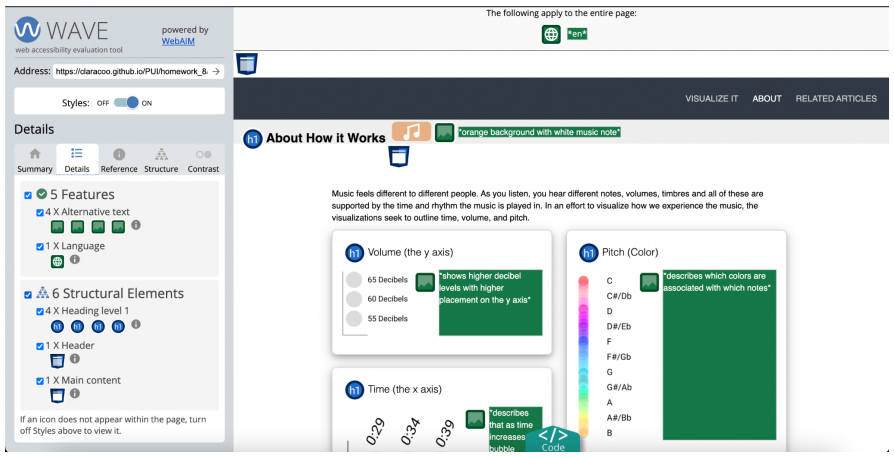
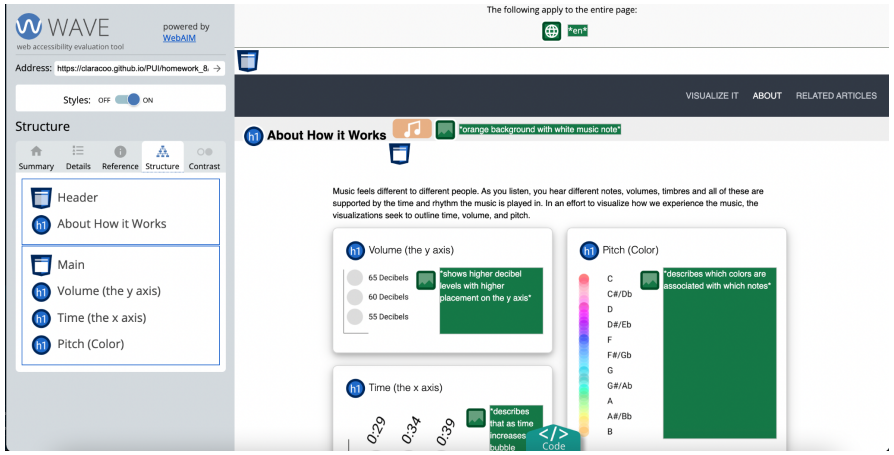
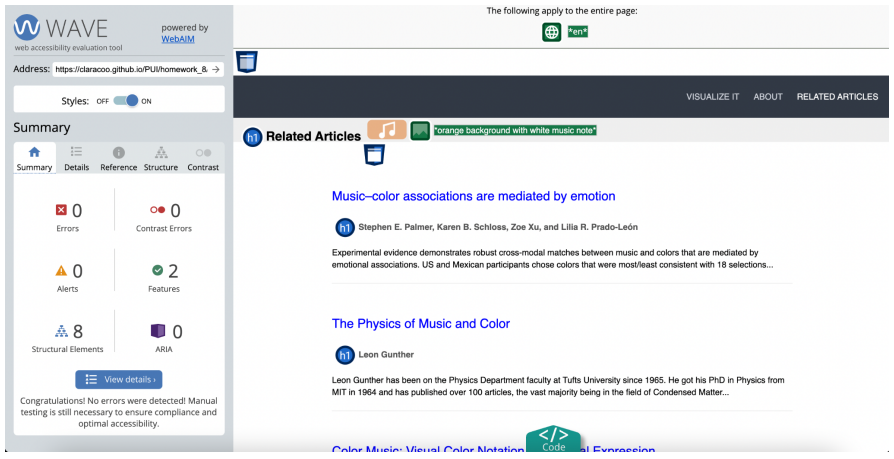
Backup Github Pages Deployed Link:
https://claracoo.github.io/PUI/homework_8/HTML/home.html

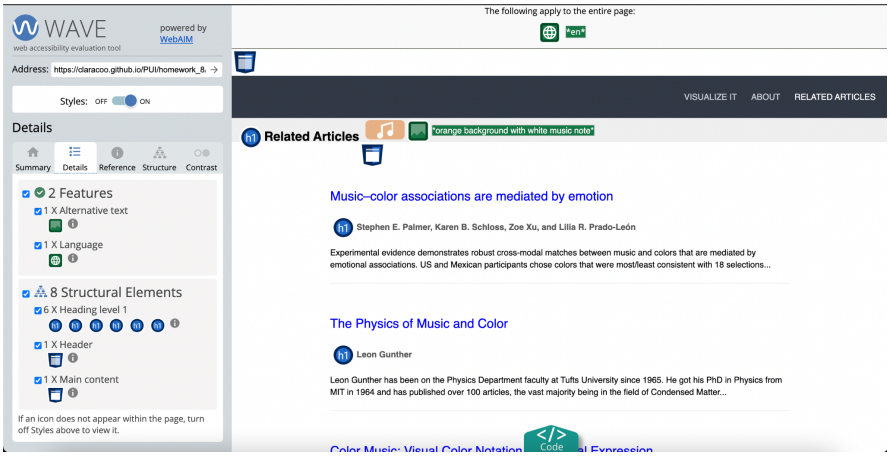
Youtube Link:
https://youtu.be/xoIOaT_VVPQ

WAV Screen Shots:

home.html	summary	

	details	
	structure	
about.html	summary	

	details	
	structure	
related.html	summary	

	details	
	structure