Rāgā

The Music Visualization Tool

Part 1: Website Description

As a Carnatic vocalist, I have often found it difficult to memorize the positions of musical notes in a rendition while learning to sing it for the first time. This was until I learned the art of complimenting music with rhythm - moving my hands to the beat to register small intricacies that make classical music stand out. By combining audio and visual cues, learning classical music can be made easier. Through my website Rāga, singers can easily visualize the notes and melodies of Carnatic classical music through color, shape, and movement. The target audience for the website is comprised of novice singers and music enthusiasts who are particularly interested in learning Indian classical music. The website stores song data in the form of musical notes and path patterns that trace the transitions between the notes. It also stores audio sources for each song. This audio data in the form of frequency and amplitude is then used to render visualizations in real time. The website engages all types of users, including users with disabilities, by implementing supplemental media to learn music. This media includes visuals, audio, and text to invoke different senses. The website has been modularized to ensure high scalability, extending its use beyond the realm of Indian Classical music. Songs can be easily stored along with their corresponding notes and path progressions, and Rāgā will generate dynamic visualizations that respond to the audio. This flexibility positions Rāga as a tool for exploring a wide range of musical traditions while enhancing learning experiences for singers and music enthusiasts alike. By bridging the gap between sound and sight, Rāga offers a novel way to appreciate and learn the beauty of classical music.

Part 2: User Interaction

- **Select a rāgā to visualize**. Click on the side arrows on the Rāgā Catalog page to navigate to the rāgā of your choice; then click on the cover image to select it.
- **Listen to each note**. Click on the Play control in the audio player at the bottom of the Path Visualizer page to listen to the sound of every note.

- **Progress to various notes**. Click on the side arrows on the Path Visualizer page to navigate to the note of your choice.
- **Play/pause the sound of each note**. Click on the Play/Pause control in the audio player at the bottom of the page to either play or pause the audio accompanying each note.
- **Visualize the song**. Click on the cyan-colored button labeled “Visualize the Song” on the Path Visualizer page to proceed to the consequent webpage displaying dynamic song visualizations.
- **Play/pause the visualizations**. Click on the cyan-colored button labeled “Play”/”Pause” at the bottom of the Frequency Visualizer page to toggle between playing and pausing the audio and visualization.

Part 3: External Tool

1. **Swiper.js**
    - Swiper.js had readily implemented accessibility features such as ARIA labels. It also enabled me to incorporate keyboard control to move between different options in the carousel.
    - I used Swiper.js to create an album cover flow effect for the rāgā catalog on the landing page. Additionally, I used it to create a carousel effect while selecting the note audio and animation to play on the Path Visualizer page.
    - It provides me with an instance of the Swiper class that I can modify and customize. This helped me to create an album cover flow effect and carousel effect while ensuring accessibility.
2. **P5.js**
    - P5.js helped me explore creative coding in the realm of 2D visualizations. Through P5.sound, it also incorporates the audio data into its visualizations which gave me more flexibility by using a single library.
    - I used P5.js to record the frequency and amplitude values of different audio sources. I then mapped these data values to create a waveform with varying thicknesses.
    - This helped me add dynamism to the visuals on my website and make it responsive to the window’s width and height. It also ensures accessibility for users with disabilities through its describe() and describeElement() functions.

Part 4: Design Iteration

During FP2, I designed two screens on Figma intending to test them to understand which information is critical to aid target users' learning process and determine if the labeling of the interface is intuitive for new users. The first screen encouraged users to explore the music catalog, successfully achieving its purpose through intuitive buttons and card interactions. The second screen was, however, static after the first interaction, confusing the users about the next possible steps and their interpretation of the information presented. In an attempt to improve the design before implementation, I added navigation buttons to move between musical notes and pages to enhance the users' sense of freedom and control. Emphasizing accessibility, I decided not to implement a drag-the-ball along-the-path interaction on the Path Visualizer page. I implemented a responsive carousel effect for selecting options on both the landing page (Rāgā Catalog) and the Path Visualizer page. In this iteration, I went beyond my initial scope and implemented a P5.js canvas for visualizing music for augmenting hearing-impaired users.

Part 5: Implementation Challenge

- Initially, I started the project by hardcoding musical notations and path values in the HTML code. As the semester progressed, I learned more about Javascript, and this knowledge helped me write more modular and cleaner code that significantly enhanced reusability.
- While trying to implement sound functionality in P5.js, I struggled to find community resources for debugging. This prompted me to reference the documentation on the P5.js website to understand the workings of each predefined function, especially for ensuring accessibility.

Part 6: Generative AI Use

Usage Experiences by Project Aspects

- *Usage*: [Yes/No]
- *Productivity*: [1-Much Reduced, 2-Reduced, 3-Slightly Reduced, 4-Not Reduced nor Improved, 5-Slightly Improved, 6-Improved, 7-Much Improved]

Tool Name	Ratings	design	plan	write code	debug
Gemini	Usage	No	No	Yes	Yes
Gemini	Productivity	4	4	5	7

Usage Reflection

Impact on my design and plan

- It matched my expectations and plan in FP2 in that Gemini helped me quickly sift through a number of Javascript libraries and frameworks to suggest the most appropriate one for the context of my project.
- It matched my expectations and plan in FP2 in that I did not use Gemini to create curves and path progressions between various musical notes since it did not allow me to customize them and ignored minute nuances.
- It matched my expectations and plan in FP2 in that I did not use Gemini to generate standalone ideas to implement within the project.
- It did not match my expectations and plan in FP2 in that Gemini helped me write code to navigate the integration of P5.js while dynamically updating DOM elements with the help of event listeners. It also helped me with suggested values for CSS properties to make the website responsive for smaller screen sizes.
- Gemini did not influence my final design and implementation plan because it does not provide customization or desired flexibility in terms of setting up a schedule that is open to unexpected changes. Specifically, I found myself improvising my proposed plan for the PUI project depending on challenges encountered on the learning curve for using a specific framework/library. Another drawback was limited access to the Premium version of Gemini. Only after purchasing the Premium plan close to the end of the Fall semester was I able to get close to the desired results from Gemini after it referenced the history of prompts.

Use patterns

- I accepted the generations when Gemini suggested libraries and frameworks followed by a justification of proposed use cases in the context of my project.

- I critiqued/evaluated the generated suggestions when Gemini tried to remove all forms of hardcoded values during debugging. This was particularly hard to evaluate when I had hardcoded SVG values for individual path progressions for each rāgā/song.
- Additionally, I critiqued Gemini's functionality when it required me to reupload project files to debug the code after I manually made changes to it.

Pros and cons of using GenAI tools

- **Pros**
    i. Gemini helps identify repeated patterns in code across files.
    ii. Gemini helps inculcate good coding habits for easier maintainability.
- **Cons**
    i. Gemini can oversimplify and overgeneralize customized code like creative art forms in libraries like P5.js.
    ii. Gemini reinforces automation bias that prevents programmers from reevaluating the model's generated output relative to our context because we assume that the machine has produced the correct result.

The usage log is present at the following link: Gemini

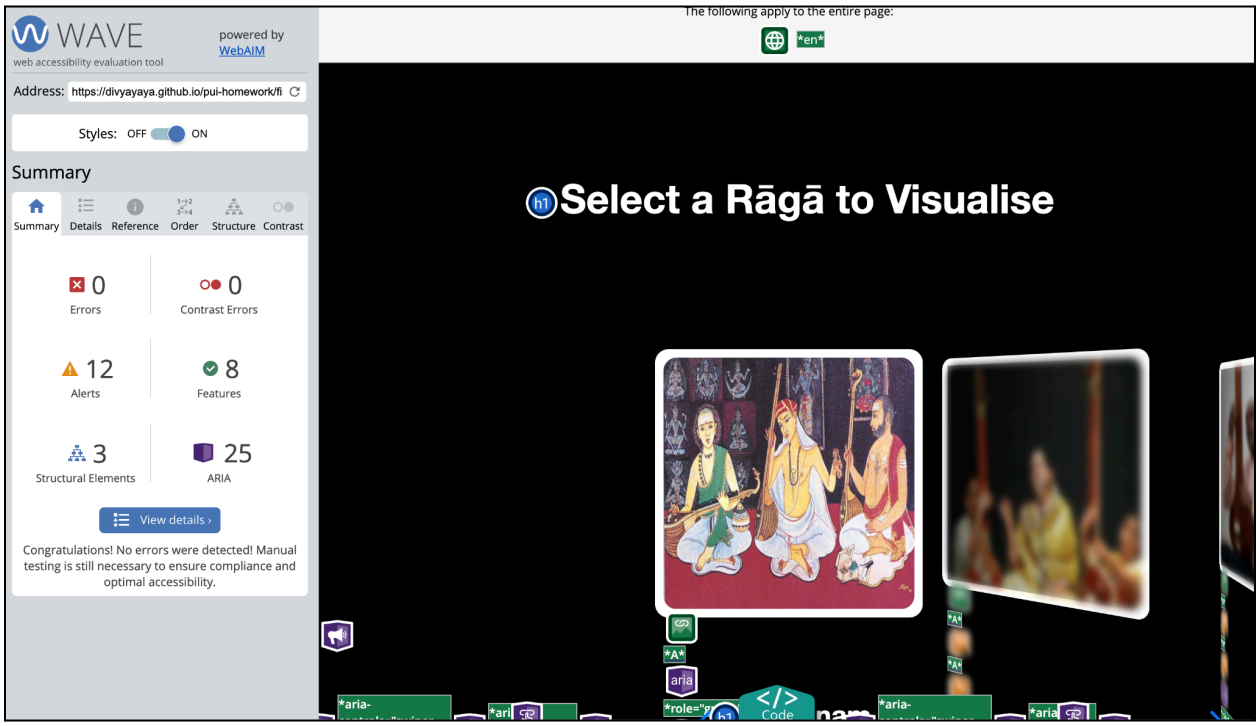
Part 7: Testing

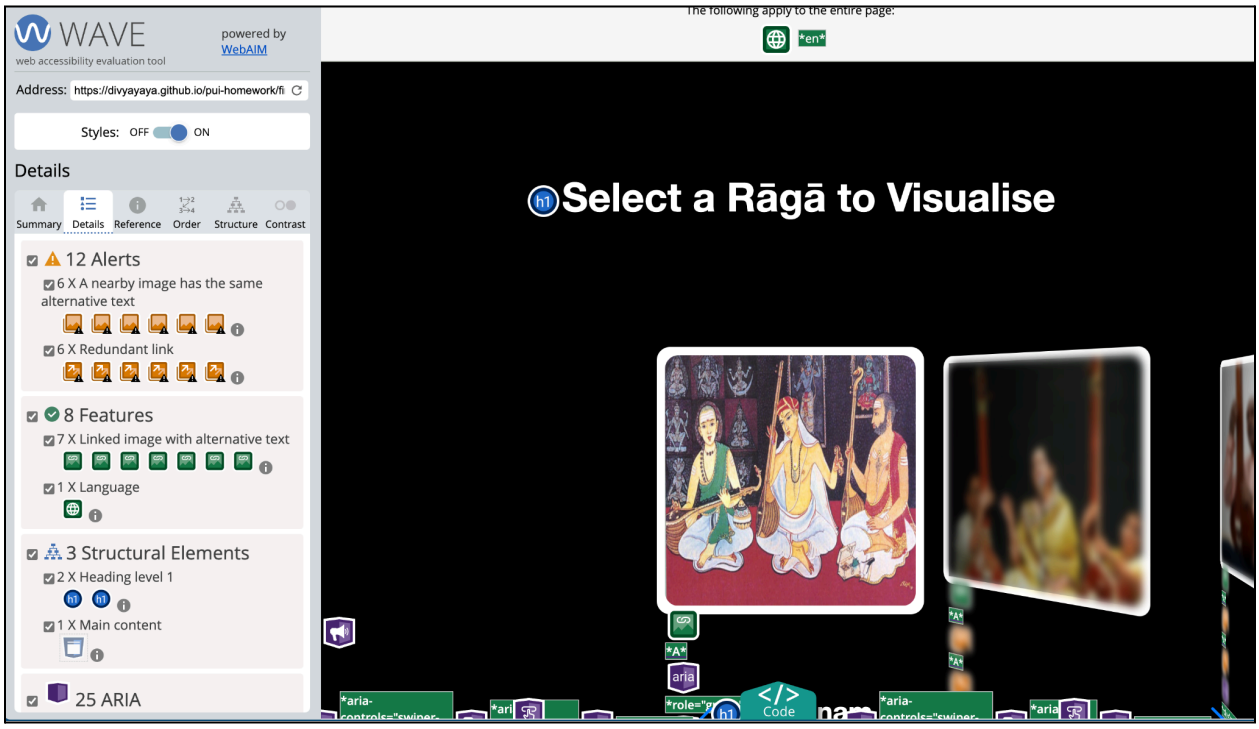
Please test the website with the following screen sizes:

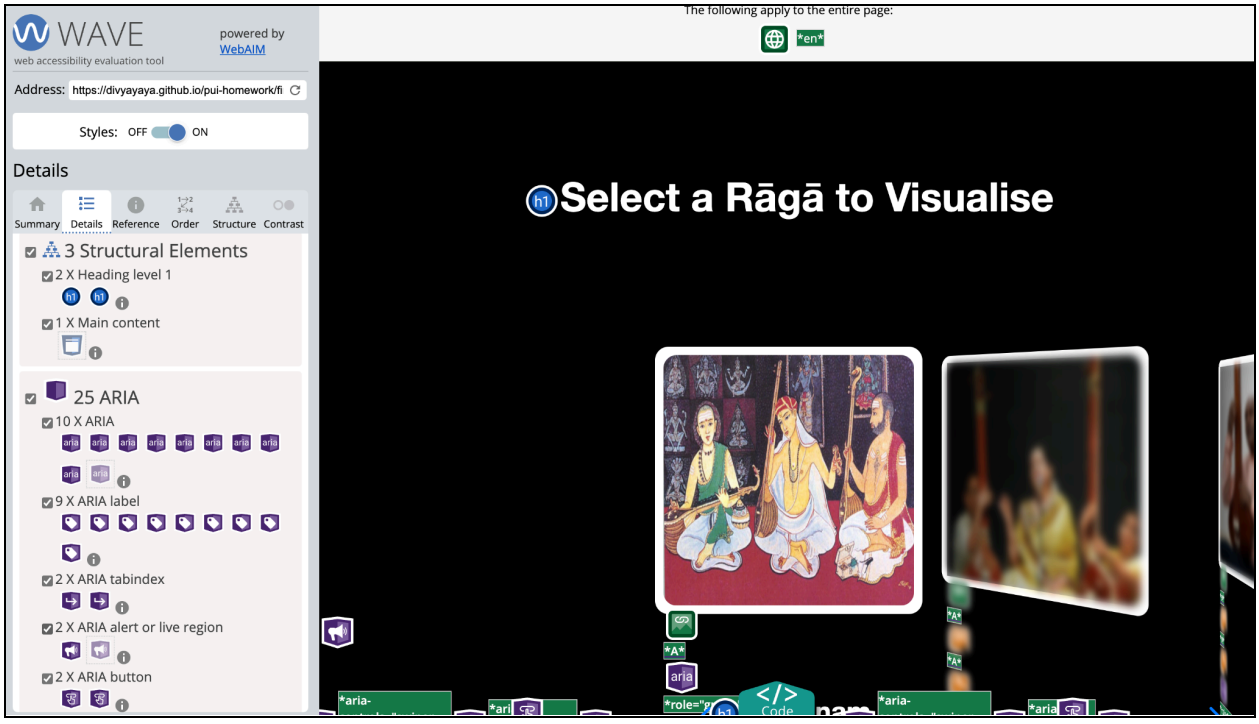
- 1280 x 720 pixels
- iPad Air (width: 820 pixels)

Part 8: Appendix

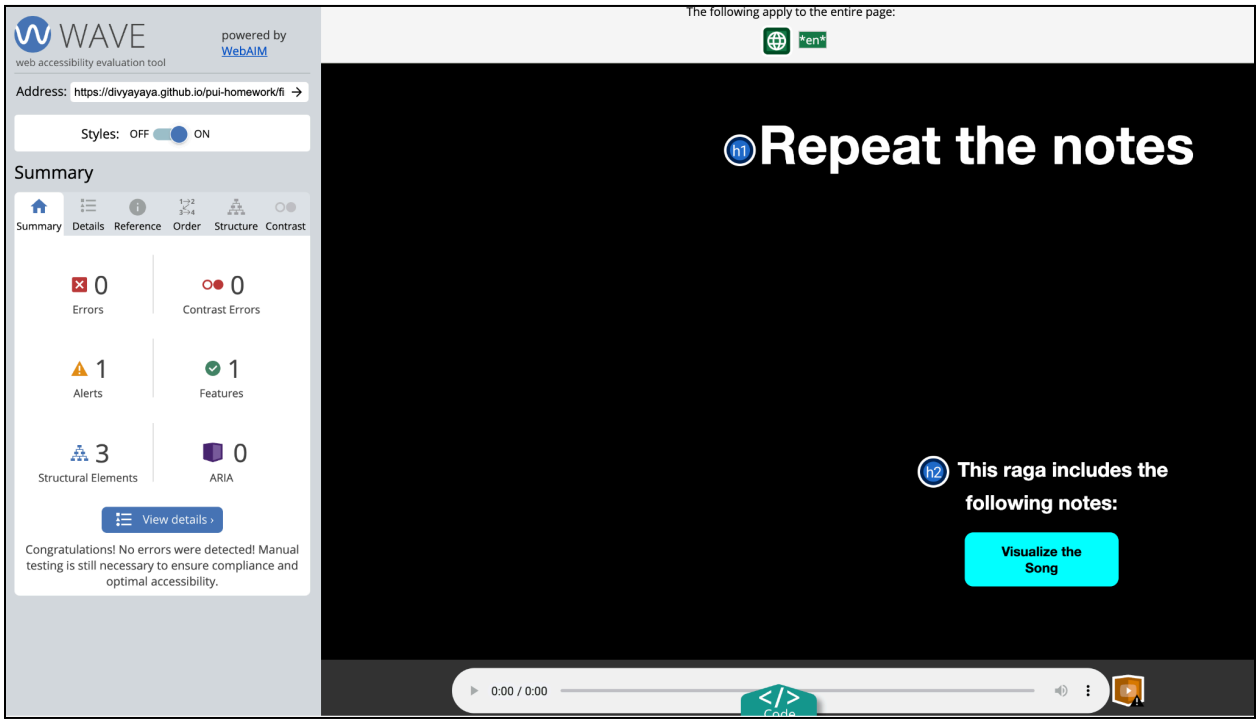
(1) “Rāgā Catalog” page

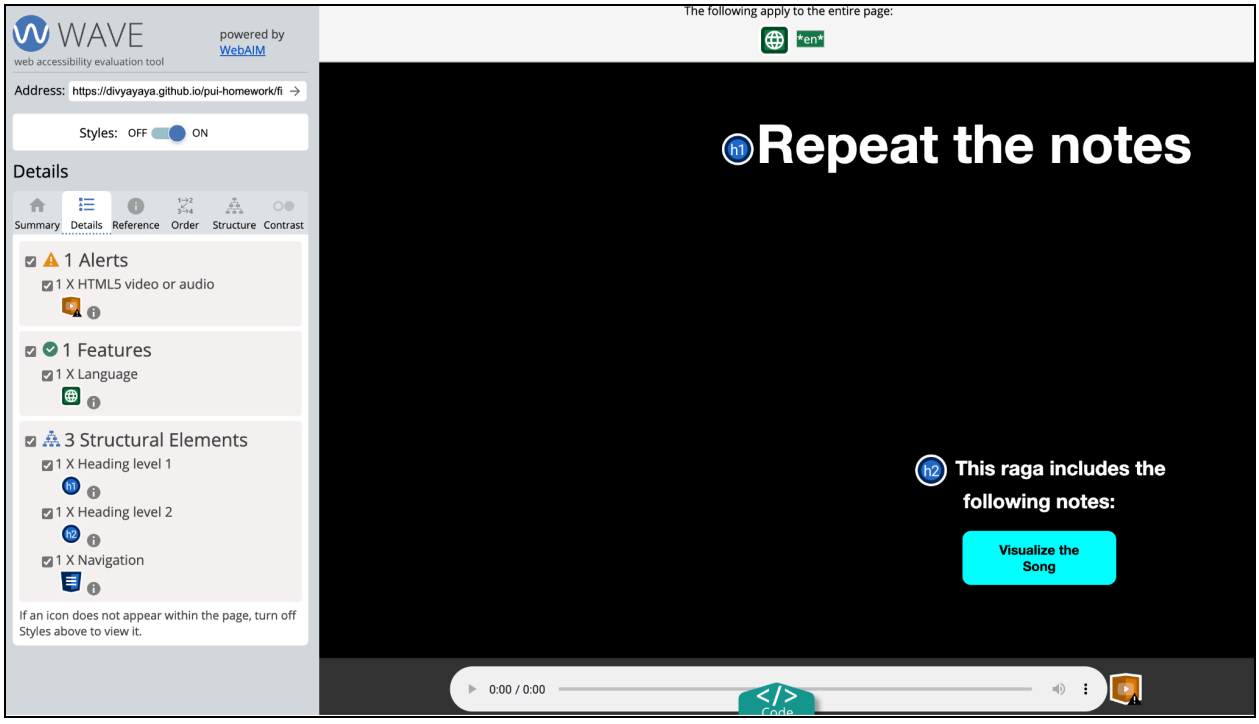


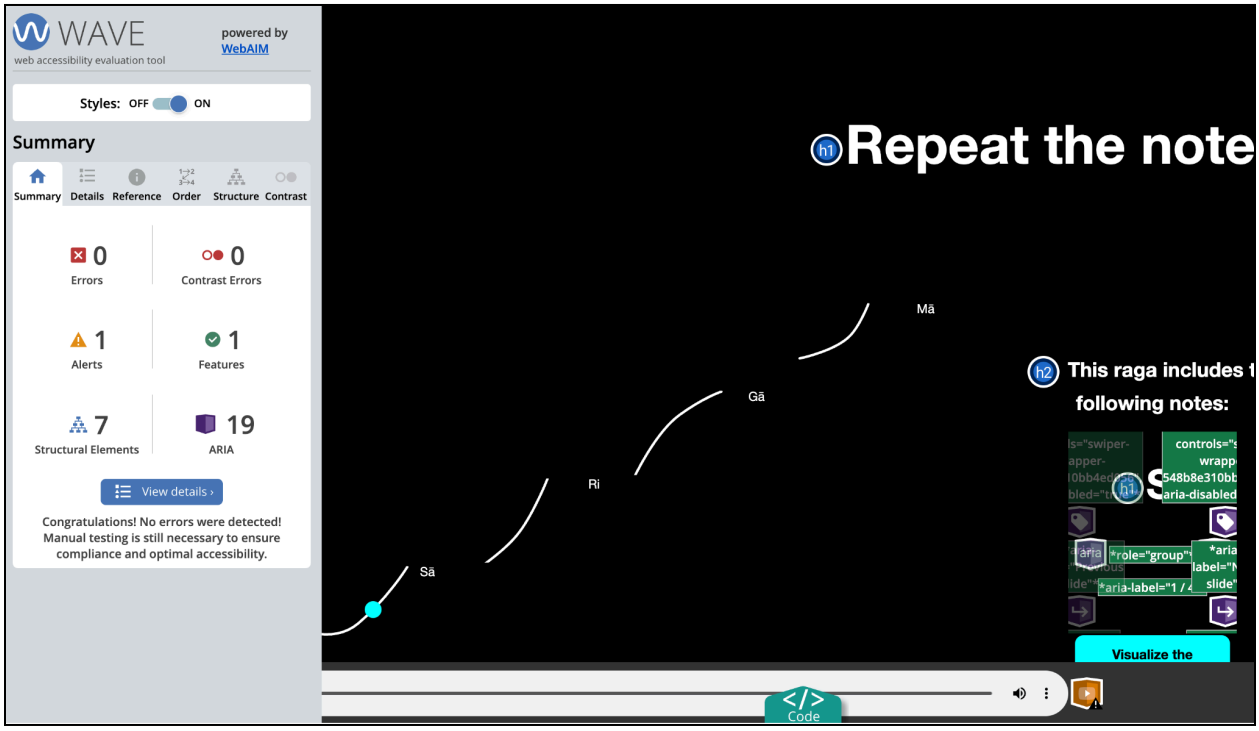


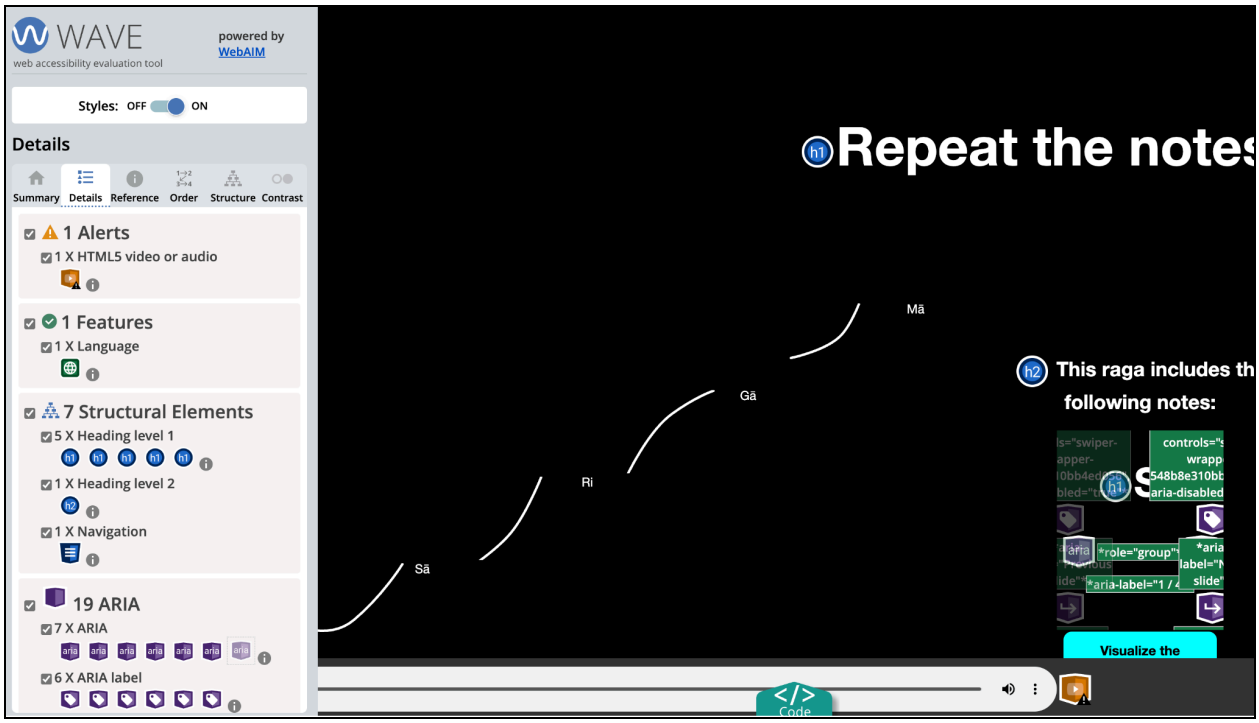


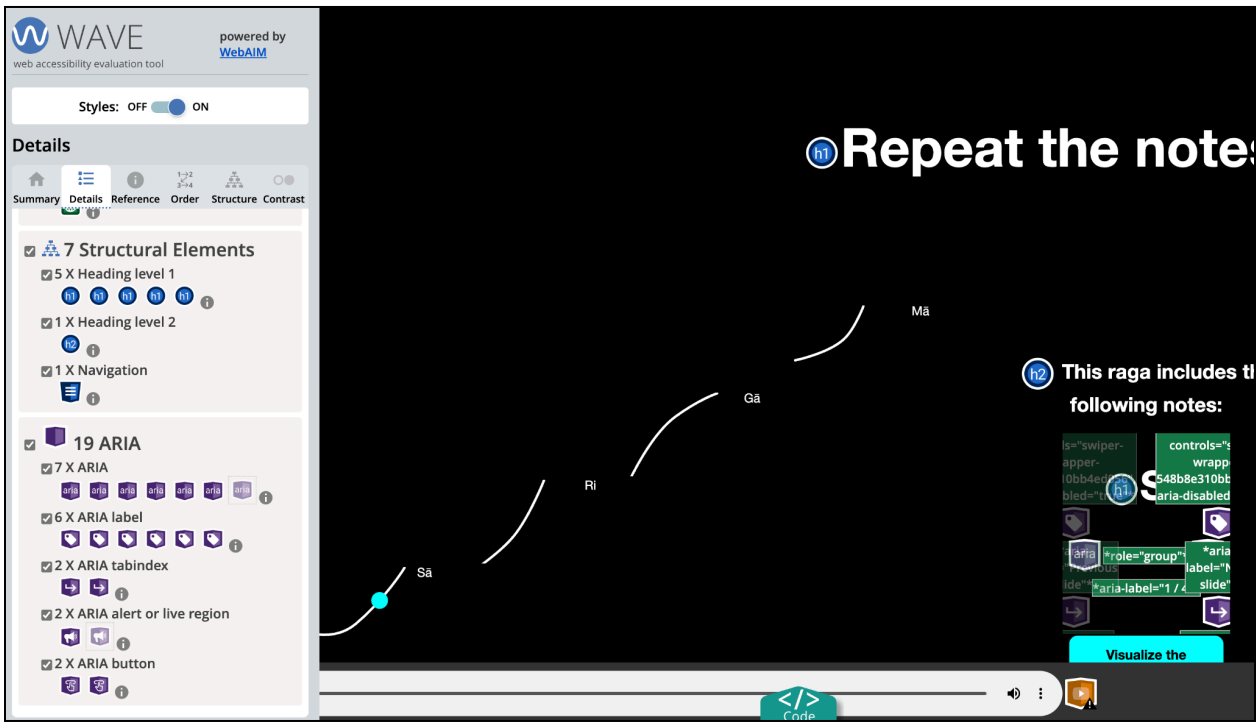
(2) “Path Visualizer” page



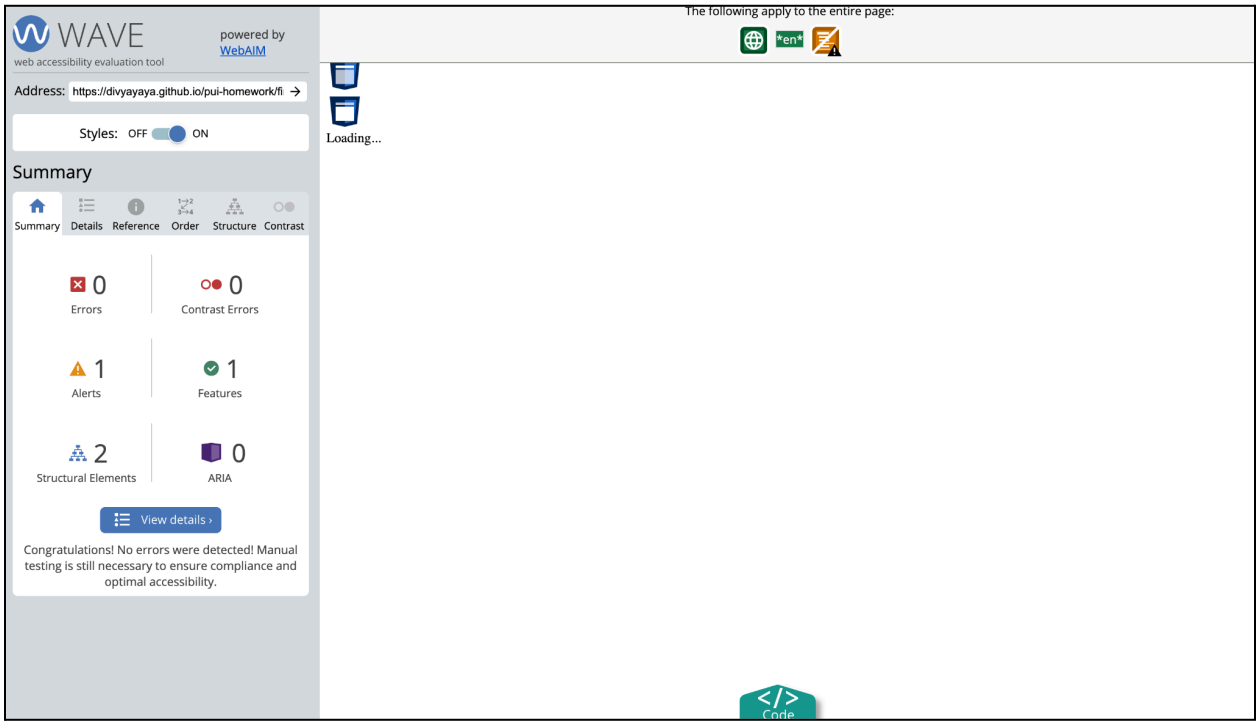

WAVE
web accessibility evaluation tool
powered by WebAIM
Address: https://divyayaya.github.io/pui-homework/fi
Styles: OFF ON
Details
Summary Details Reference Order Structure Contrast
1 Alerts
1 X HTML5 video or audio
1 Features
1 X Language
3 Structural Elements
1 X Heading level 1
1 X Heading level 2
1 X Navigation
If an icon does not appear within the page, turn off Styles above to view it.
The following apply to the entire page:
en
Repeat the notes
This raga includes the following notes:
Visualize the Song
0:00 / 0:00
Code

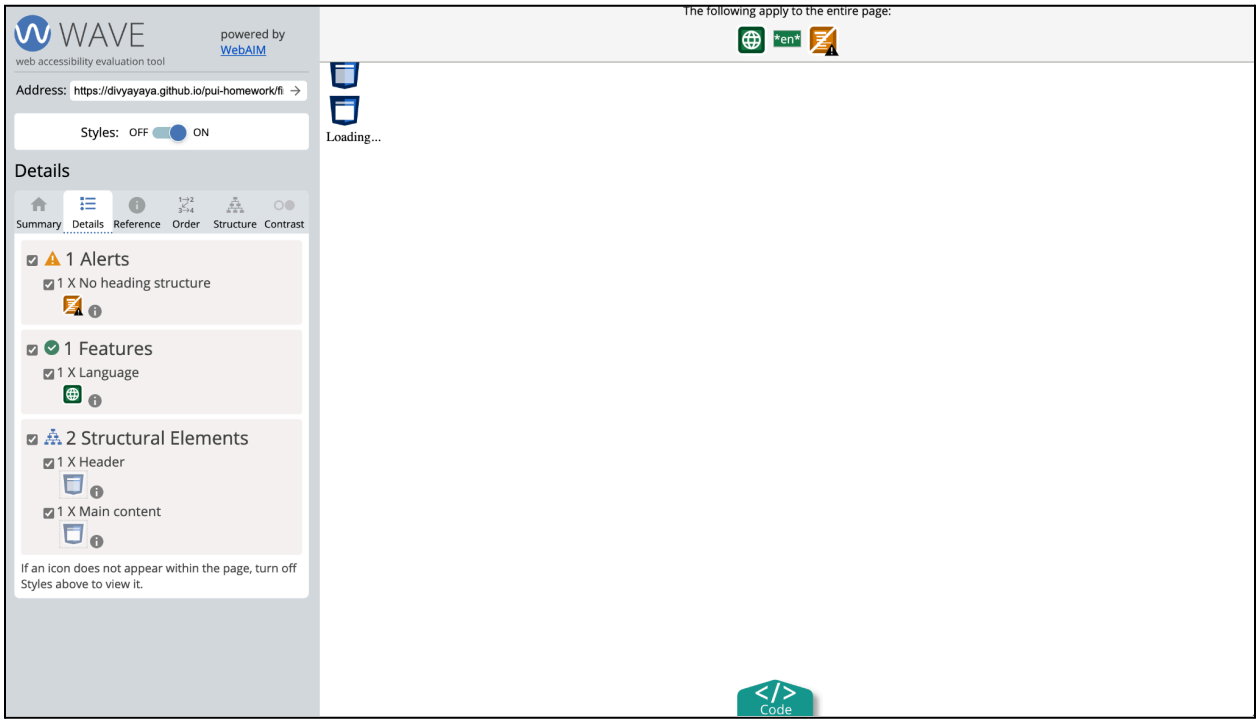

WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Summary
Summary Details Reference Order Structure Contrast
0 Errors
0 Contrast Errors
1 Alerts
1 Features
7 Structural Elements
19 ARIA
View details ›
Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.
Repeat the note
Mā
Gā
Ri
Sā
This raga includes 1
following notes:
Visualize the
Code

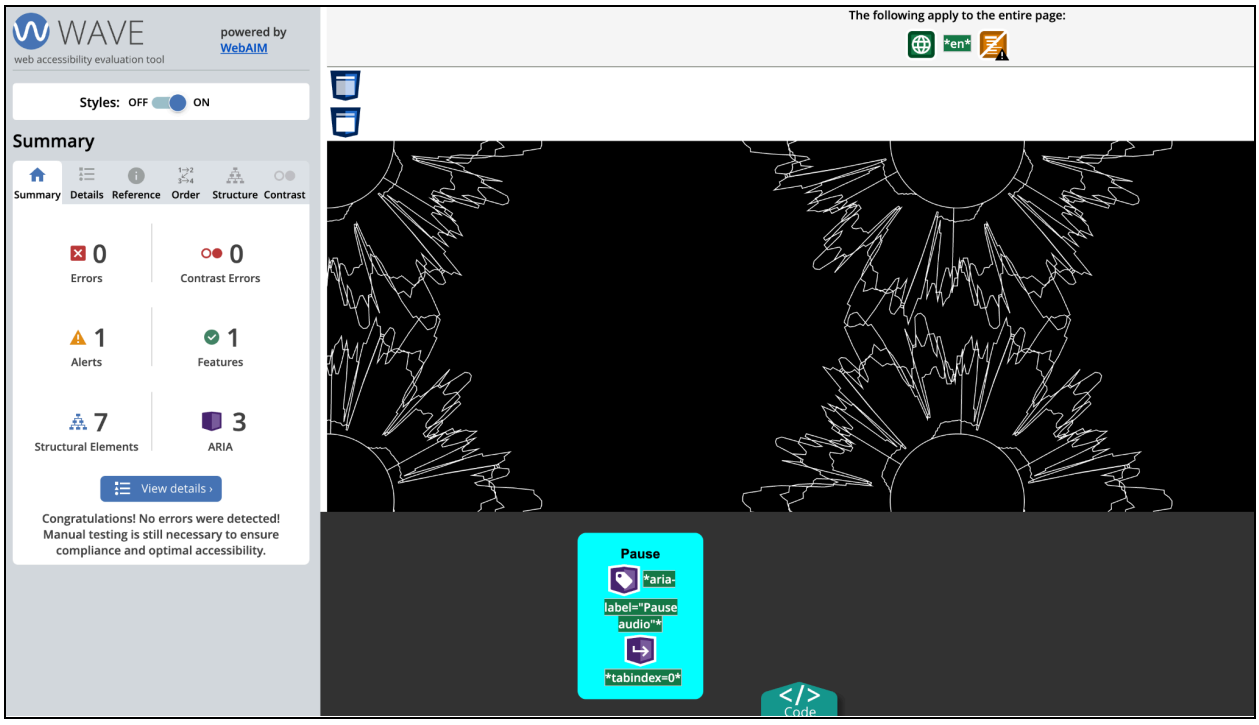

WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Details
Summary Details Reference Order Structure Contrast
1 Alerts
1 X HTML5 video or audio
1 Features
1 X Language
7 Structural Elements
5 X Heading level 1
1 X Heading level 2
1 X Navigation
19 ARIA
7 X ARIA
6 X ARIA label
Repeat the notes
Sā
Ri
Gā
Mā
This raga includes th
following notes:
Visualize the
Code

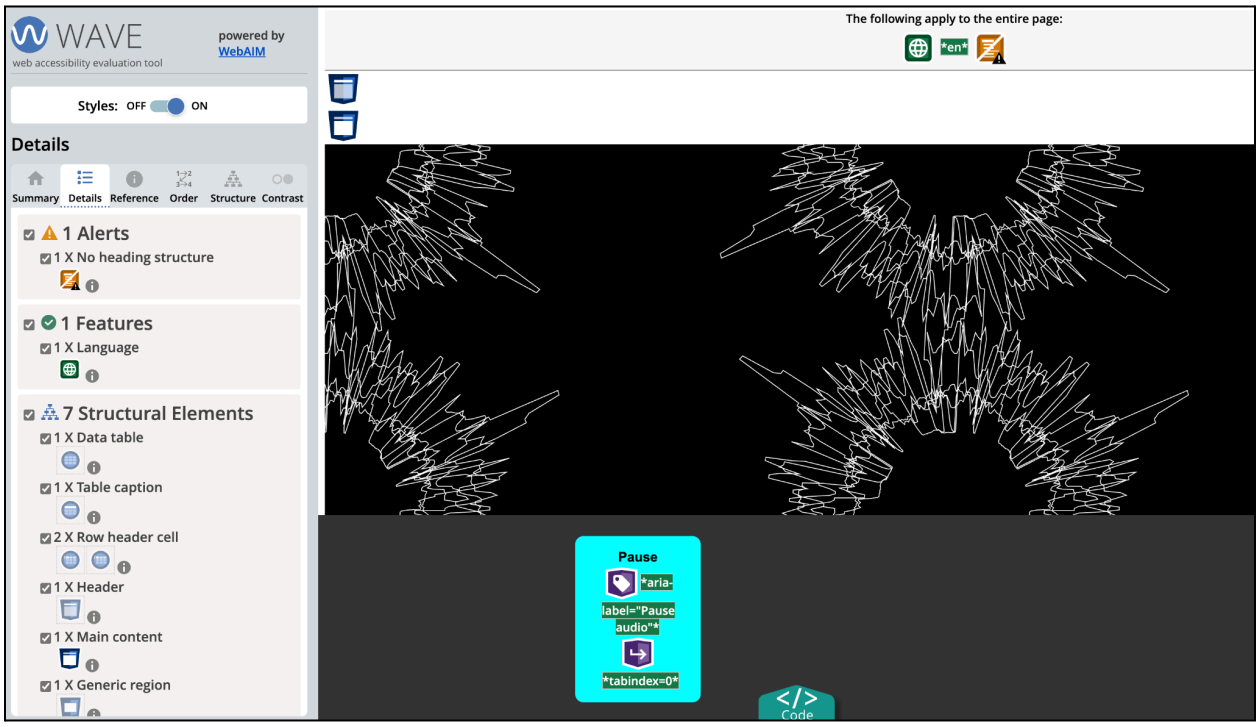

WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Details
Summary Details Reference Order Structure Contrast
7 Structural Elements
5 X Heading level 1
1 X Heading level 2
1 X Navigation
19 ARIA
7 X ARIA
6 X ARIA label
2 X ARIA tabindex
2 X ARIA alert or live region
2 X ARIA button
Repeat the notes
Sā
Ri
Gā
Mā
This raga includes th
following notes:
Visualize the
Code

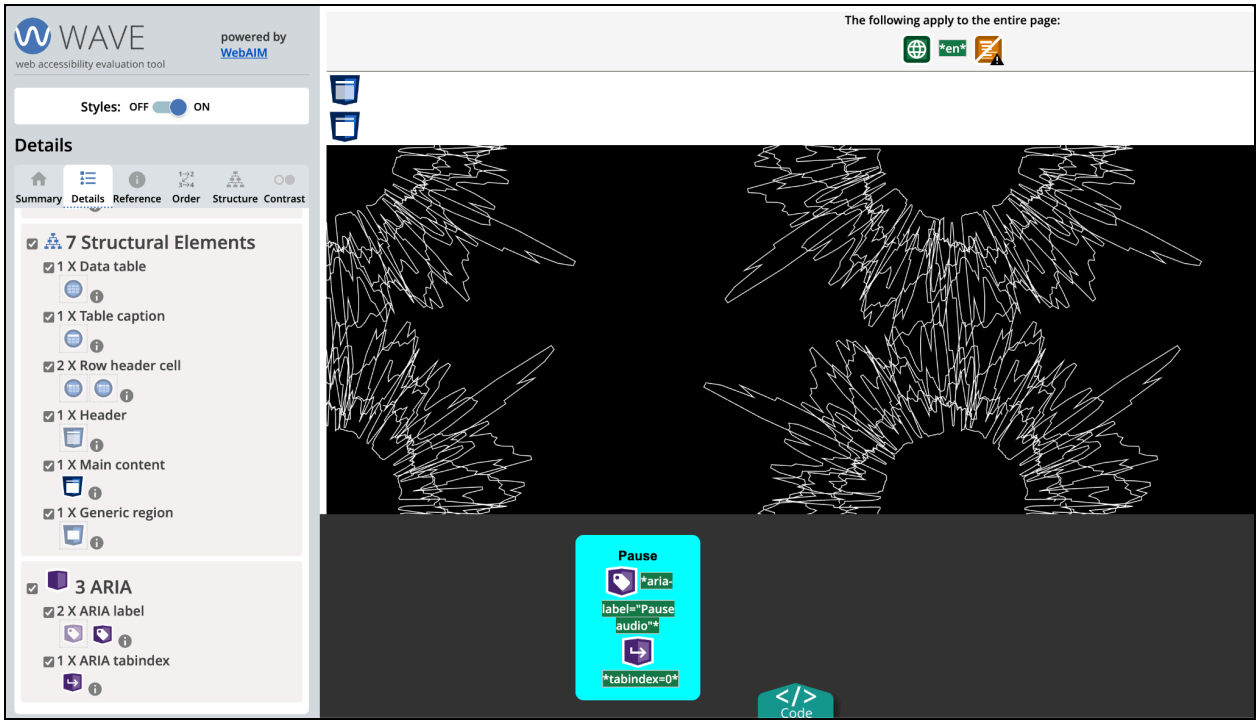
(3) “Frequency Visualizer” page






WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Summary
Summary Details Reference Order Structure Contrast
0 Errors
0 Contrast Errors
1 Alerts
1 Features
7 Structural Elements
3 ARIA
View details ›
Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.
The following apply to the entire page:
en
Pause
aria-label="Pause audio"
tabindex=0
Code


WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Details
Summary Details Reference Order Structure Contrast
1 Alerts
1 X No heading structure
1 Features
1 X Language
7 Structural Elements
1 X Data table
1 X Table caption
2 X Row header cell
1 X Header
1 X Main content
1 X Generic region
The following apply to the entire page:
en
Pause
aria-label="Pause audio"
tabindex=0
Code


WAVE
web accessibility evaluation tool
powered by WebAIM
Styles: OFF ON
Details
Summary Details Reference Order Structure Contrast
7 Structural Elements
1 X Data table
1 X Table caption
2 X Row header cell
1 X Header
1 X Main content
1 X Generic region
3 ARIA
2 X ARIA label
1 X ARIA tabindex
The following apply to the entire page:
en
Pause
aria-label="Pause audio"
tabindex=0
Code