

# 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 Rāgā, 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 Rāgā 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, Rāgā 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: Testing

Please test the website with the following screen sizes:

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

## Part 7: Appendix

### (1) “Rāgā Catalog” page

The following apply to the entire page:

powered by WebAIM

Address: <https://divyayaya.github.io/pui-homework/fi/>

Styles: OFF ON

**Summary**

Errors: 0 Contrast Errors: 0

Alerts: 12 Features: 8

Structural Elements: 3 ARIA: 25

[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:

powered by WebAIM

Address: <https://divyayaya.github.io/pui-homework/fi/>

Styles: OFF ON

**Details**

12 Alerts

- 6 X A nearby image has the same alternative text
- 6 X Redundant link

8 Features

- 7 X Linked image with alternative text
- 1 X Language

3 Structural Elements

- 2 X Heading level 1
- 1 X Main content

25 ARIA

**Select a Rāgā to Visualise**

The following apply to the entire page:

WAVE  
web accessibility evaluation tool  
powered by WebAIM  
Address: <https://divyayaya.github.io/pui-homework/fi>

Styles: OFF ON

Details

3 Structural Elements  
 2 X Heading level 1  
 1 X Main content

25 ARIA  
 10 X ARIA  
 9 X ARIA label  
 2 X ARIA tabindex  
 2 X ARIA alert or live region  
 2 X ARIA button

## (2) "Path Visualizer" page

The following apply to the entire page:

WAVE  
web accessibility evaluation tool  
powered by WebAIM  
Address: <https://divyayaya.github.io/pui-homework/fi>

Styles: OFF ON

Summary

Errors: 0 | Contrast Errors: 0  
Alerts: 1 | Features: 1  
Structural Elements: 3 | ARIA: 0

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

View details >

h1 Select a Rāgā to Visualise

h2 This raga includes the following notes:

Visualize the Song

0:00 / 0:00

The following apply to the entire page:

**Details**

- Alerts:** 1 (1 X HTML5 video or audio)
- Features:** 1 (1 X Language)
- 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.

**Repeat the notes**

**This raga includes the following notes:**

**Visualize the Song**

Code: </>

**Summary**

Errors	Contrast Errors
0	0

Alerts	Features
1	1

Structural Elements	ARIA
7	19

Congratulations! No errors were detected!  
Manual testing is still necessary to ensure compliance and optimal accessibility.

**Repeat the note**

**This raga includes 1 following notes:**

Sā, Ri, Gā, Ma

Code: </>

**WAVE**  
powered by WebAIM

Styles: OFF ON

**Details**

- Alerts:** 1 X HTML5 video or audio
- Features:** 1 X Language
- Structural Elements:**
  - 5 X Heading level 1
  - 1 X Heading level 2
  - 1 X Navigation
- ARIA:** 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:**

This raga includes the following notes:

Visualize the code

**WAVE**  
powered by WebAIM

Styles: OFF ON

**Details**

- Structural Elements:**
  - 5 X Heading level 1
  - 1 X Heading level 2
  - 1 X Navigation
- ARIA:** 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:**

This raga includes the following notes:

Visualize the code

(3) “Frequency Visualizer” page

The following apply to the entire page:

Address: <https://divyayaya.github.io/pui-homework/fi>

Styles: OFF  ON

Summary

Category	Count
Errors	0
Contrast Errors	0
Alerts	1
Features	1
Structural Elements	2
ARIA	0

[View details >](#)

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

</> Code

The following apply to the entire page:

Address: <https://divyayaya.github.io/pui-homework/fi>

Styles: OFF  ON

Details

▲ 1 Alerts

- 1 X No heading structure

● 1 Features

- 1 X Language

▲ 2 Structural Elements

- 1 X Header
- 1 X Main content

If an icon does not appear within the page, turn off Styles above to view it.

</> Code

The following apply to the entire page:

Styles: OFF ON

**Summary**

Errors: 0 Contrast Errors: 0

Alerts: 1 Features: 1

Structural Elements: 7 ARIA: 3

Congratulations! No errors were detected!  
Manual testing is still necessary to ensure compliance and optimal accessibility.

View details >

Pause \*aria-\*  
label="Pause"  
audio-\*  
\*tabindex=0\*

</> Code

The following apply to the entire page:

Styles: OFF ON

**Details**

Alerts: 1 X No heading structure

Features: 1 X Language

Structural Elements: 7

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

Pause \*aria-\*  
label="Pause"  
audio-\*  
\*tabindex=0\*

</> Code

The following apply to the entire page:

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

Pause  
label="Pause"  
audio\*\*  
+tabindex=0\*

</>  
Code

This screenshot shows the WAVE web accessibility evaluation tool interface. The left sidebar displays a tree view of the page's structure with 7 structural elements and 3 ARIA annotations. The main content area shows a complex, abstract graphic composed of many thin, jagged white lines on a black background. A callout box highlights a specific ARIA annotation for a button labeled "Pause". The annotation details are: label="Pause", audio\*\*, and +tabindex=0\*. At the bottom right, there is a code editor window with the text '</> Code'.