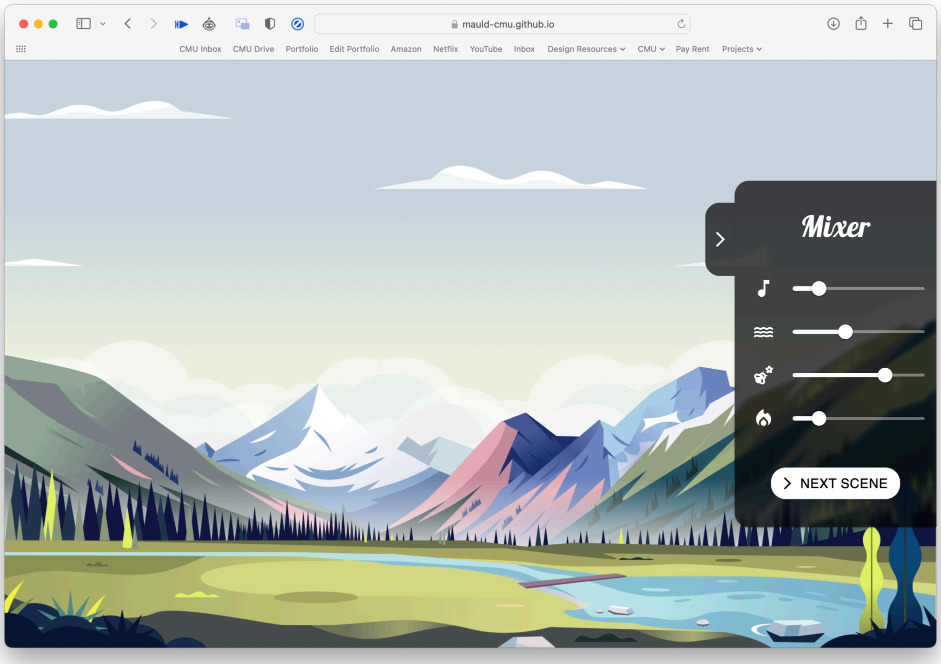
Part 1: Description



Bring Me *Vibes* is an ambient music and soundscape website, paired with a beautifully animated scene. A user can customize the individual volumes of environmental sounds to meet their exact needs.

I designed with the intention of making it as easy and flexible as possible to create new soundscapes and scenes. The process of adding a new scene is entirely dynamic and data driven - all a developer needs to do is add a new data structure and files to the site's code and then the application will do all the heavy lifting of building out a scene, animating it, and creating mixing components. Here is all that is needed to create a new dynamic scene:

- An entry in the "sceneData" object in "SceneData.jsx"
- A list of sounds/songs and their file paths (as well as names & icons to represent them)
- A background gradient or color
- A list of image layers with the following properties
  - Z-height: for proper layering
  - Motion: is it moving or not?
  - Speed: how fast is it moving?

For this site, I wanted a fun and unique way to create a peaceful backdrop to day-to-day life, like studying, cooking, or just hanging out. I hope this web app will be useful for people who like doing things with music in the background, but also want some nice visual interest to go along with it.

This app works best on Tablet and Desktop Interfaces, and works okay on mobile devices (the dynamic background will appear compressed, otherwise entirely functional).

Check it Out: https://mauld-cmu.github.io/bring-me-vibes/
Warning: Some browsers block auto-play. To fix this, trigger the next scene or click on each of the slider sound icons to start playing.

Part 2: Interaction

- **Hide Mixing Sidebar**: click the ‘>’ button to the left of the Mixer to hide it, featuring a slick animation!
- **Show Mixing Sidebar**: click the ‘<’ button to show the Mixer again.
- **Sound Sliders**: click & drag the sound sliders in the mixer to adjust the volume of individual effects or music
- **Play & Pause individual Sounds**: click the icons to the left of the sliders to play or pause the specific sound
    - This is sort of a hidden action, mostly for debugging. However, it can be useful in cases where browsers disable autoplay. The same effect can be achieved by a user simply sliding the volume to 0.
- **Next Scene**: click the next scene button to see a new scene/soundscape arrangement.
    - Since it takes quite a while to create the drawings for scenes, this app currently only supports 2. I do plan on adding more, however!

Part 3: Tools Used

React

Why: I chose React so that I could easily manage state throughout the many distinct components in my application. Many of the components in my app need to communicate with and update each other, and React made that much simpler to achieve.

How: I installed it with npm and utilized functional components.

What it Adds: React enabled me to create stateful components that interacted with each other efficiently, reducing code complexity and enabling faster development.

React MUI

Why: I wanted to use React MUI so that I could utilize efficient design patterns for common components, like buttons, sliders, and cards. No need to reinvent the wheel!

How: I installed it with npm and imported them into my React components where needed.

What it Adds: Though there were times I regretted using material UI because of the immense amount of re-styling I had to do, I ultimately enjoyed the simplicity of not having to re-invent common inputs like sliders in React. It ultimately made my design more usable with little added effort.

React Audio Player

Why: As it turns out, controlling audio playback with plain HTML and JS is extremely convoluted. This react wrapper for the HTML audio tag simplified that process greatly, offering me better control over my audio files.

How: I installed it with npm and imported an Audio Player component into each of my Slider components.

What it Adds: It makes the controlling of audio much simpler for me, ultimately creating a better experience for the user.

GH-Pages

Why: This library drastically simplified the process of launching a react page on GitHub Pages

How: I followed the helpful instructions we used for homework 1-6

What it Adds: An easy build & launch process for my site onto GitHub pages. Without it, I doubt I would have figured out how to get that page to launch!

Part 4: Iterations

I started out this project a bit more ambitious, hoping to integrate Spotify APIs and complex scenes with different weather and time of day effects. And though I still hope I will be able to add these features in the future, I realized pretty early that I had drastically over-scoped for the amount of development time I had. I had to scope back.

I got rid of my ideas for weather & time of day features, but made sure to keep the scene animations. I was originally going to work with the Spotify API, but in the end, I found it difficult enough to work with audio files stored on the app itself, so I had to scope that back too.

However, I did not want to compromise the quality of the code I wrote to crank out features. I made sure to emphasize modularity in my code design, ensuring the addition of scenes could be seamless, and that I could leave room for all the features I originally scoped for. In the end, I think I made a robust, expandable application that I can be proud to work on further in the future.

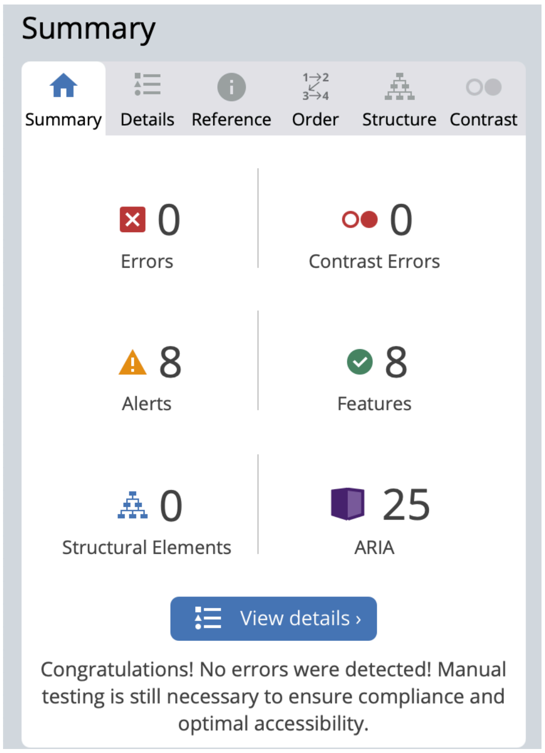
Part 5: Challenges

Figuring out React MUI components was more difficult than I expected! As it turns out React MUI components are quite prescriptive, and making changes to style or functionality can take a lot more time than expected.

Also, working with audio was a challenge. I spent nearly an entire day trying to figure out a custom audio player, and ultimately gave up and handed over the reigns to a library.

Finally, the custom animations were a challenge. For something that I thought would be relatively simple (tiling & animating an image), I went through several design iterations with CSS animations before finally getting it to work.

Appendix: Accessibility Report



Summary

Summary Details Reference Order Structure Contrast

0 Errors

0 Contrast Errors

8 Alerts

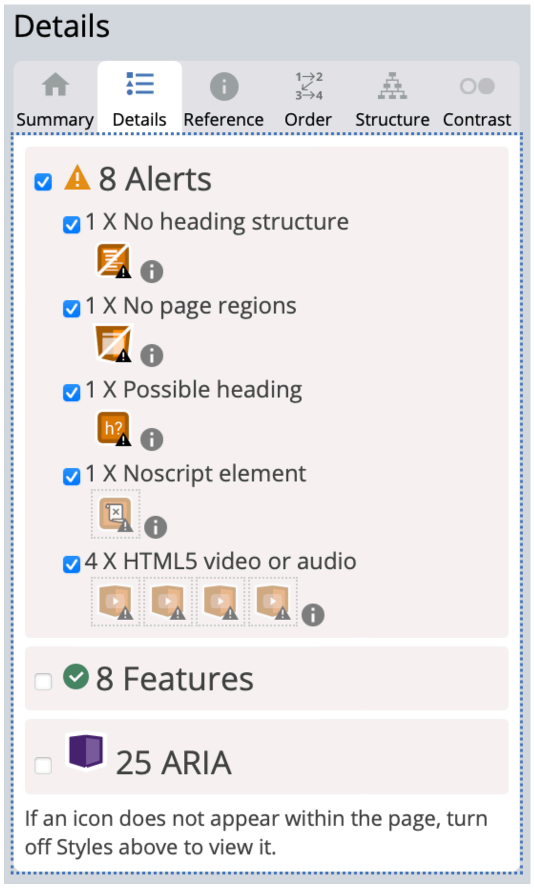
8 Features

0 Structural Elements

25 ARIA

View details ›

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



Details

Summary Details Reference Order Structure Contrast

8 Alerts

1 X No heading structure

1 X No page regions

1 X Possible heading

1 X Noscript element

4 X HTML5 video or audio

8 Features

25 ARIA

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

Addressing Warnings:

- **No Heading Structure**: The only text content in the entire app is the word “Mixer,” and it is defined as a “title” property in the Card.
- **No Regions**: This project has no header, footer, or nav components, it is entirely content.
- **Noscript Element**: The noscript element is built-in to React to warn the user when they have JS disabled
- **HTML Audio**: HTML audio elements all have text descriptions, sound effects cannot have transcriptions