

Part 1

This project is an interactive artistic visualization of sentiment associated with various texts, aimed to communicate the diversity of language in and nuances of emotional expression. The variety of texts also engages users with an interesting glimpse into the wide range of experiences that cause people to feel certain emotions, ultimately conveying the richness of the human emotional experience. Users are able to read through texts associated with different emotions, which may provide a cathartic effect - if someone is sad, they may want to look at happy texts to remind them of the happier moments in life or look at sad texts to be reminded that they are not alone. The target audience is the general population/adults, but I assume that young adults will be the most drawn to it given the use of slang and emojis. The texts will be pulled from an existing Kaggle dataset

(https://www.kaggle.com/datasets/ishantjuyal/emotions-in-text?select=Emotion_final.csv)

that contains snippets of text with their associated sentiment. Given that all the texts from the dataset were presented in lowercase, I decided to make the entire website lowercase to improve consistency and aesthetics. Additionally, I incorporated some fun elements into the interface such as displaying emojis on hover, bouncing navbar buttons, and vivid colors in order to keep the user engaged and add an element of delight to the website.

Part 2

- Hover popup: Hover over each line of text to see the emotion associated with it, as well as the text highlighted in the color associated with that emotion (sad = blue, anger = red, fear = purple, surprise = orange, love = pink, happiness = yellow)
- Filtering: Click on any emotion listed in the top navbar to filter by emotion (e.g., to view only happy texts, click "Happiness" on the top nav bar)
 - Hovering over navbar items will make them bounce
- Refresh: Refresh the page to randomize the text that is displayed on the homepage

Part 3

- I used Bootstrap Navbar to make the top navbar responsive and accessible from various devices/screen sizes. I used it by linking and adding the documentation provided to my HTML file, customizing and replacing the placeholder items, and styling it through CSS.
- I used Animate.css to make the navbar items bounce when hovered over. This not only adds a fun element to the page, but also communicates interactivity in that the user knows that they can click the navbar items. I used it by linking it to my HTML file and then adding the documentation provided to my CSS file.

- I used the `Kursor.js` library to change the cursor to a circle in order to make the website more delightful and aesthetically pleasing. This helps make the website unique and more engaging for the user. I used it by linking and adding the documentation provided to my HTML file.

Part 4

I originally designed my website in a newspaper style, where all the texts were continuously displayed in a block of text resembling a newspaper with horizontal scrolling. I decided to split up the texts line-by-line and enlarge them in order to improve readability/accessibility. After realizing that color was not a sufficient signal for emotion, I added labels for the emotion on hover. I added a background color to the navbar items when focused on instead of underlined in order to make the color-coding more apparent after receiving feedback that the colors in the underlines were too subtle. Rather than displaying all the text items on the homepage, I limited it to 20 items that would randomize every time the page was refreshed so that there would be a new variety of text every time.

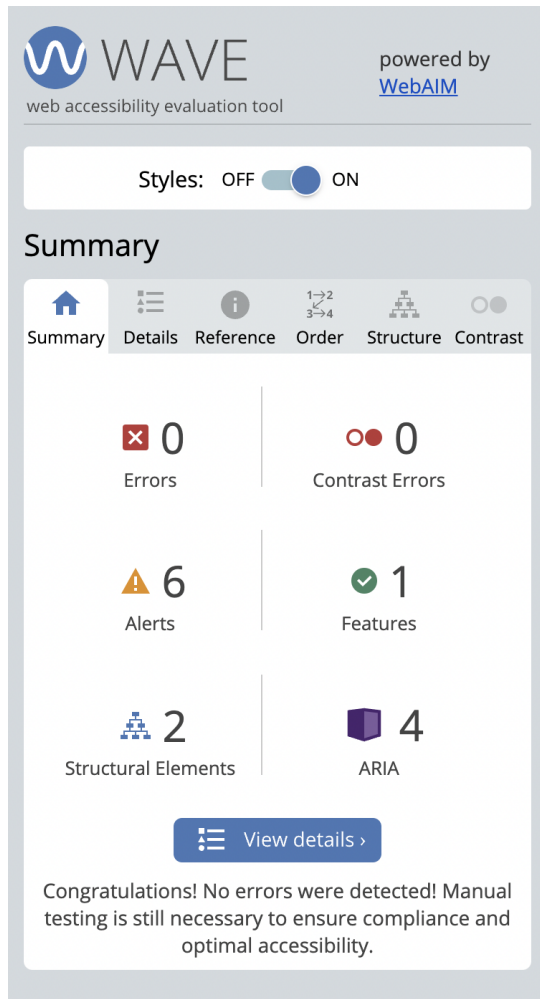
Part 5

Initially, I struggled with adding the hover functionality and filtering by emotion, and I spent significant time learning how to implement event listeners for hover and functions/for loops to filter by the emotion (key) from the JSON file. Making the website responsive was challenging - I originally used media queries, but swapped it out for Bootstrap to support a smoother responsive experience by collapsing the navbar items into a hamburger menu on smaller screen sizes. After using Bootstrap, I had some trouble with the alignment of the navbar with the items on the page. I also was not sure how to display only 20 items per page rather than the entire list, so had to learn how to use the `Math` object to limit and randomize the texts.

Responsiveness

Test the website on desktop (1440), tablet (820), and iPhone (390)

WAVE Screenshots









WAVE web accessibility evaluation tool powered by WebAIM

Styles: OFF ☐ ON ☒

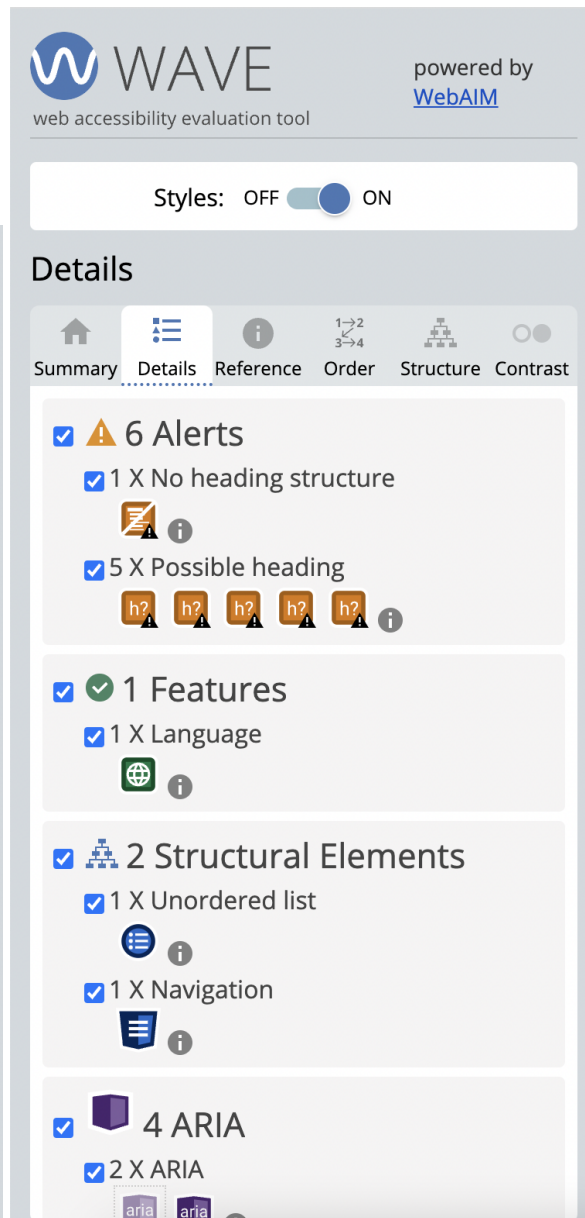
Summary

Summary Details Reference Order Structure Contrast

 0 Errors	 0 Contrast Errors
 6 Alerts	 1 Features
 2 Structural Elements	 4 ARIA

[View details >](#)

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
















WAVE web accessibility evaluation tool powered by WebAIM

Styles: OFF ☐ ON ☒

Details

Summary Details Reference Order Structure Contrast

- ☒  6 Alerts
 - ☒ 1 X No heading structure
 - 
 - ☒ 5 X Possible heading
 -     
- ☒  1 Features
 - ☒ 1 X Language
 - 
- ☒  2 Structural Elements
 - ☒ 1 X Unordered list
 - 
 - ☒ 1 X Navigation
 - 
- ☒  4 ARIA
 - ☒ 2 X ARIA
 - 