Archana Ramakrishnnan – FP4 Final project

Programming Usable Interfaces, Fall 2022

Part 1: Description of Website

TODO: Grow is a planner and productivity app. It serves as a reminder to the user of tasks they need to work through a to-do list view. Its unique selling point is that it shows a visual description of your progress! Users will see a tree that slowly grows based on how much work you finish. This motivates the user to keep going and primes them to protect and grow their tree friend and protect their progress from distractions. This positive reinforcement will serve as a healthy way to achieve your goals. Most importantly, there is also a to-do list for self care in pink. If the user creates and finishes self care tasks for themselves, the tree blooms with pretty pink flowers. The target audience is anyone who wants to be determined to work hard and make progress. It could especially appeal to young adults who would like to share their tree on social media with friends.

The tree that is rendered on the screen follows a recursive algorithm. This gives it a symmetrical and ordered look with some degree of randomness in terms of branch length and leaf/petal colors to make it look natural. The to do list has features to add, delete and check off items. Additionally, they can be filtered based on which tasks have been completed, which are active and a default option to see all items. The app shows today's date and a small prompt at the bottom to push the user to get started on building their tree. The app has elements of React accessibility embedded in it, hoping to reach users of all kinds.

URL: https://archanaramakrishnan.github.io/todo-grow/

Part 2: User interaction

Interaction types

- Form
- Checkbox
- Text input
- Buttons
- Visual feedback through the tree

Instructions

Can be tested on regular laptop, iPad Air (820x1180) and phone devices

- Click on the input box on the "work" list and add a productive to do either using the enter key or the add button
  - Add up to 3-5 productive tasks
  - Check one or two off the list
  - You will notice lush green leaves on your tree
- Add a task or two under the "self" list and one or more off. You will notice pink flowers have grown on your tree
- You will find a small instructional/motivation prompt if you scroll all the way to the bottom

Part 3: External tools

- **Name of tool**
  - P5.js
  - Material UI components
  - React.js
- **Why I chose it**
  - P5 had an online editor which I could experiment on
  - It had many references and tutorials online that were intuitive to follow and looked visually appealing
- **How I used it**
  - I had to use a ReactP5Wrapper to draw a p5 sketch in a component and then render that to the React app
  - I used a recursive algorithm to render the tree with a branch length and variables to control how many flowers would bloom
- **What does it add**
  - It adds a captivating graphic element to an otherwise tedious to do list
  - It gives immediate feedback and gratification to users that something has been checked off of their list!

These follow along tutorials greatly helped me learn and produce quality work

1. React todo list by MDN
2. Recursive tree in p5 YouTube tutorial

Part 4: Iterations

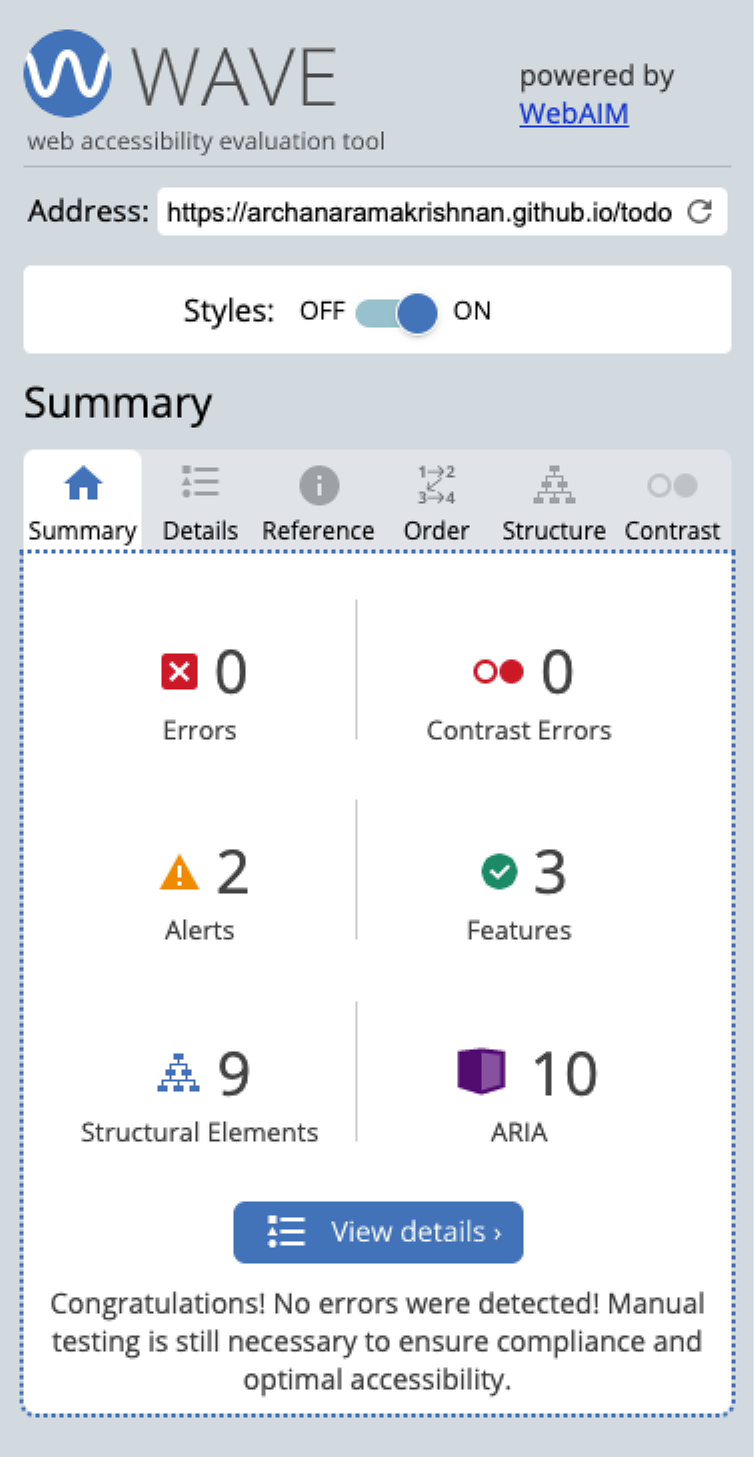
- I started off with purely the logic of the todo app and then the tree
- I only had one list (productivity) at first and focussed on rendering the leaves, after which I added the additional list and background
- At first, I had three columns with two lists and one tree, but a critique comment from a peer mentioned that I could make the tree the background and put the cards on top, which made it look much more elegant!
- Styling all the elements to feel consistent, cohesive and pleasant was my final step!

Part 5: Challenges

Some challenges I had was with positioning the blue sky gradient and the tree simultaneously on the canvas- with some translation, I was able to overcome this. Understanding how the recursion would work with showing leaves and petals was hard, but I came up with an algorithm. I wanted to make the p5 part of my project accessible as well, however I found out that the p5 accessibility library is not compatible with MacOS. In the future I will try to implement a fully accessible tree in the browser editor and then transfer it on to my development environment on my Mac.

Accessibility Report

Screenshots included below



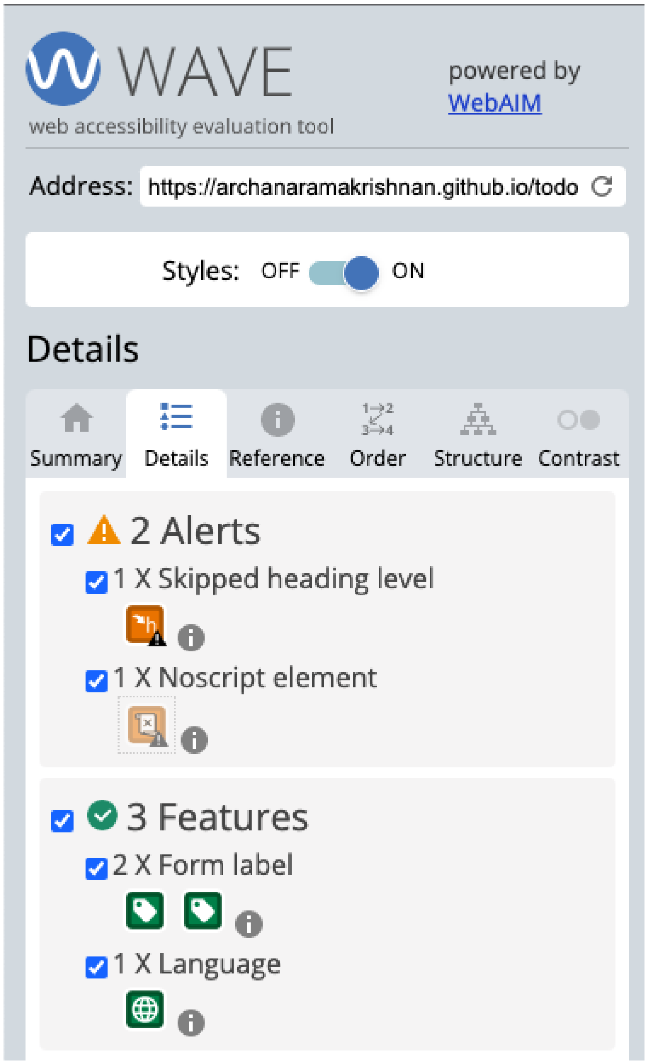
Address: https://archanaramakrishnan.github.io/todo

Styles: OFF ● ON

Summary

☒ 0 Errors	0 Contrast Errors
⚠ 2 Alerts	✓ 3 Features
9 Structural Elements	10 ARIA

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



Address: https://archanaramakrishnan.github.io/todo

Styles: OFF ● ON

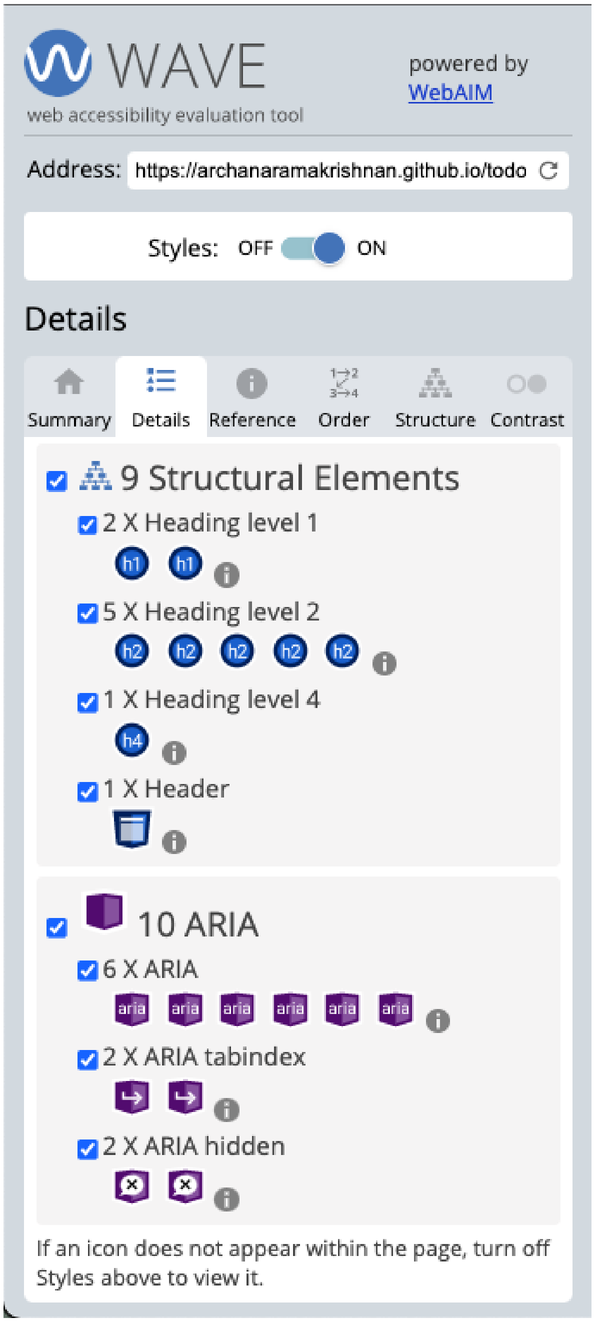
Details

☑ 2 Alerts

- ☑ 1 X Skipped heading level
- ☑ 1 X Noscript element

☑ 3 Features

- ☑ 2 X Form label
- ☑ 1 X Language



Address: https://archanaramakrishnan.github.io/todo

Styles: OFF ● ON

Details

☑ 9 Structural Elements

- ☑ 2 X Heading level 1
- ☑ 5 X Heading level 2
- ☑ 1 X Heading level 4
- ☑ 1 X Header

☑ 10 ARIA

- ☑ 6 X ARIA
- ☑ 2 X ARIA tabindex
- ☑ 2 X ARIA hidden

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