IT5039

Client-side Development

Project Document

Interactive Web Application

Submitted by

Naomi Tomasson

Student ID: 20240768

Date: Aug 18, 2024

Whitecliffe

Contents

Project Overview

The Magic 8 Ball game is a simple web application that will tell your fortunes. First, you ask a question out loud or in your head and click the "Ask me a question" button, the ball will then shake and show a response to your question.

Features and Functionalities

1. **The "Ask me a question" button** has an event listener applied to it and a random selector to change the image. When clicked the following behaviour sequence occurs.

   First, the "q-button" element is retrieved by ID from the DOM and assigned a new variable name "shakebtn". This is then added as an event listener that will occur when the element is clicked.

   Then the button is disabled to stop multiple clicks (the button was self-clicking twice the first time every time the page was opened).

   The CSS animation "shake" is first removed to reset the existing animation.

   Then a force reflow to make sure the animation gets reapplied

   Shake class gets added back to trigger animation.

   A time-out is set to remove the shake class after 1 second

   Then our "changeBallImg" (see below) function is called to randomly change the image for an answer.

   Disable is now false to enable the shake button to be used again.

2. "changeBallImg" function. This function selects an image at random out of an array and displays the new ball image to provide an answer.

   First, we make a variable to generate a random index from 0 to the length of our image array (array found at the top of JS script – 'rotationImages'). This will select a random image from the array.

   Then, finally the source of the ball image element us changed to the randomly selected image.

User Interface Design Principles applied

Design Principles for the Magic 8 Ball Application

Contrast: The design features a dark purple background contrasted with light pink text, ensuring that key elements like the header, image, and buttons stand out. This not only enhances readability but also draws attention to the most important parts of the interface.

Sizing and Visual Hierarchy: The 8 ball, as the focal point of the application, is the largest element on the page, commanding the user's attention. The button sizes are thoughtfully adjusted to guide the user's focus, particularly to the question button, ensuring a clear and intuitive user flow.

Spacing: Adequate spacing is applied between elements to create a clean, aesthetic layout. This careful arrangement not only enhances the visual appeal but also makes the interface more navigable, as each element is distinct and easy to find.

Colour: A palette of playful purples was selected to evoke a sense of magic and mystery, perfectly aligning with the theme of the Magic 8 Ball. The use of contrasting colours on buttons further differentiates the elements, making for easier navigation.

Consistency: A consistent colour scheme, limited to three harmonious shades, is maintained throughout the interface. This consistency creates a cohesive and polished look, ensuring that all elements complement each other.

Typography: The chosen typeface embodies a magical, fantasy-inspired theme, reinforcing the playful and mysterious personality of the game. Visual hierarchy is achieved through the use of two font sizes: a prominent, large heading and smaller, readable text on the buttons. The typeface not only enhances the thematic feel but also ensures clarity and ease of reading.

Web Accessibility Principles

I used the accessibility feature in developer tools to check for issues, in contrast, keyboard and text issues and there were no issues detected. Everything is up to standards.

Alt attribute: I have added alternative text to the attributes for labelling.

Landmarks: Header & main sections are landmarked/defined in the HTML to provide AT users with information about the page’s overall layout.

Heading numbering: I used the h1 element for my main/only heading, headings are important when trying to get information across in the right order of importance.

Tabindex: I did not need to add this as the focusable elements to use on the page are only a play and reset button that fall in the natural HTML focus order.

Styling: I set focus styling in CSS to help visibility when navigating. I added styling to the buttons so the user can see what button is selected.

Text visibility: All text stands out so the user can read it easily.

Text Alternatives: I have applied text alternative attributes to all HTML elements to make the page readable.

ARIA: I have added ARIA labels to the heading and buttons.

There wasn't too much required to add for ARIA as the controls are straightforward for this web application. Using the tab and enter keys allows you to play the game just as easily as if someone were to use a mouse. Following the rules of ARIA use I decided to not add many ARIA controls to my HTML elements as my research informed me that using them would be unnecessary since the semantics and behaviours required to make the application accessible were built into the native HTML. I found that adding any extra ARIA controls may cause more errors than assistance.

Testing and Debugging

Keyboard navigation was tested to confirm that all interactive elements are accessible.

Feature to Test	Test Description	Expected Result	Pass/Fail
Ask me a question button	Test if question button runs correctly on both mice click and enter button.	Click button, ball image will perform animation and then change and display image.	Pass
Reset button	Test if reset button runs correctly on both mice click and enter button.	Click button, ball image will reset back to original image with 8 symbols.	Pass

Conclusion

I've realized that my biggest challenge often lies in organizing my approach to writing code. To address this, I've recently started creating a step-by-step list of instructions as we have done in lessons, so that I may follow them, ensuring I achieve the desired results more efficiently.

The most significant difficulty I encountered with this project was implementing the shake animation in JavaScript. It took several attempts to get it right, particularly by adding and removing the disabling feature on the ball. Initially, the ball wouldn't shake on the first click of the "q-button," although it worked fine afterwards on the following clicks. And then it started double-clicking on the first click. Adjusting the code to ensure the animation didn't repeat unnecessarily was quite challenging. I was surprised at how long it took to write such a small piece of code, because of the research/information I had to dig through..

One aspect I particularly enjoyed was learning about accessibility. It introduced me to concepts I hadn't previously considered, making this part of the project both educational and enjoyable.

Overall, while the concept and application seemed straightforward, tackling these issues for the first time proved to be a learning curve. I'm confident that with experience, similar projects will be easier to handle in the future.

References

:focus-visible. (n.d.). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/CSS/:focus-visible

Animation. (n.d.). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/CSS/animation

Button that refreshes the page on click. (n.d.). Stack Overflow. https://stackoverflow.com/questions/29884654/button-that-refreshes-the-page-on-click

Content structure | web.dev. (n.d.). web.dev. https://web.dev/learn/accessibility/structure?continue=https%3A%2F%2Fweb.dev%2Flearn%2Faccessibility%23article-https%3A%2F%2Fweb.dev%2Flearn%2Faccessibility%2Fstructure

How do I delay a function call for 5 seconds? (n.d.). Stack Overflow. https://stackoverflow.com/questions/4738595/how-do-i-delay-a-function-call-for-5-seconds

How to add an image file to an object of an array in JavaScript ? (2024, February 12). GeeksforGeeks. https://www.geeksforgeeks.org/how-to-add-an-image-file-to-an-object-of-an-array-in-javascript/

How to create JavaScript delay function. (n.d.). Stack Overflow. https://stackoverflow.com/questions/16873889/how-to-create-javascript-delay-function

How to repeat shake animation on every click of a Button? Simple JS. (n.d.). Stack Overflow. https://stackoverflow.com/questions/66452961/how-to-repeat-shake-animation-on-every-click-of-a-button-simple-js

JavaScript reset. (n.d.). www.javatpoint.com. https://www.javatpoint.com/javascript-reset

Learn accessibility. (n.d.). web.dev. https://web.dev/learn/accessibility

Pasi, A. (2023, November 14). *How to design useful and usable focus indicators*. Deque. https://www.deque.com/blog/give-site-focus-tips-designing-usable-focus-indicators/

"Shake" CSS Keyframe animation. (2015, August 24). CSS-Tricks. https://css-tricks.com/snippets/css/shake-css-keyframe-animation/

W3Schools.com. (n.d.). W3Schools Online Web Tutorials. https://www.w3schools.com/cssref/css3_pr_animation-timing-function.php

(n.d.). YouTube. https://www.youtube.com/watch?v=kgqMcFrT51o