CSIS 1280 Final Project Report

By Jack Braun. ID: 300349282

Introduction:

My website is designed to help people plan their vacations. I decided to go with a blue colour for my project as it invokes the thought of the sea and beaches. My site contains three pages: a home page; a country page, which shows off why someone would want to travel to those countries; and a hotel cost calculator. The math behind the hotel cost calculator is not based on any real-life examples and is entirely fictional for the sake of showing my skills. I was interested in making a page like this as I have been wanting to go travelling for quite some time, and a fully functional website of this type would be a very helpful resource to me.

Website Layout:

Here is a layout wireframe of Page 1:



To breakdown this layout I will start with the components that are identical throughout all my projects pages. The header is comprised of a span with an H1 element located within. I used a span to center the heading text within the header. The nav is not a part of the header as I did not want it to be given a background image. The nav is also made using flexbox wrap so the content is not crushed when the page shrinks. There is then an HR element to separate the universal layout from the page specific layout.

Now onto specific layout for this page. First there is a heading which differentiates which page you are looking at. The first section is a simple about us section detailing what our website does. Following section 1 there is a heading for section 2, it is not located within section 2. I did this as section 2 is a flexbox and I did not want the heading to be involved. Section 2 is made up of two divs, each containing one image and two P elements. The section is flexboxed in a way that both divs can appear in one or in two rows. The content within the divs has been flexboxed in such a way that all content will remain in the same columns, however they will change size to match the screen when necessary. The footer is a simple HR element followed by a P containing copyright information.

Website Organization:

Each page has a header with a background image, and the name of the website. The first page has no JavaScript. The first page's purpose is to act as a home page. It has an about us section and a reviews section. The reviews section has a photograph of the reviewer, a P element containing their final rating and a brief text review.

When entering the second page you will find that it is mostly empty apart from a drop down select list. The select list has three options: USA, France and Canada. Each one of the options has an embedded YouTube link hidden in the value attribute. Upon selecting one of the options, a JavaScript event listener will be activated. The JavaScript will take the hidden value located within your selected option and place it into the src attribute. It will then find what option you have selected and place a brief text description of your selected country into a P element below the video. The purpose of the second page is to act as a tourism / showing off page of a few different countries and some interesting things you may find in them.

Lastly page 3 contains a form that depending on the options selected will calculate the hotel cost of where you want to travel. The form has 4 different inputs: Hotel Rating, which is of type number; Choose number of beds, which is a radio input; Location, which is a drop down select input; and duration of stay, which is a range. The entire form has had validation implemented. There are two separate event listeners connected to this form. The first event listener is to get what the selected duration of days is and display the value beneath the input in an empty P element.

The second event listener on page 3 is the form validation. When all inputs have good data entered, the calculateCost function is run. This function collects the data from all inputs and stores them into variables which I've labeled as data variables. Each data variable is run through a complicated nested loop, there are several arrays which will assign a mathematical value to their corresponding data variables. The loops will find the exact value of each data variable and apply their corresponding mathematical values to new variables which I've called cost variables. At the end of the loop when all cost variables have their assigned values, they are multiplied together, finally the sum of the cost variables is placed in a P element directly below the form.

Highlight:

The part I spent the most amount of time on was the calculateCost function. I had to spend a large chunk of time working out how to setup the conversion from inputted data to a pre-determined set of mathematical variables. I ended up using a 4 deep nested loop. The loop takes all four inputs, finds their corresponding mathematical value, and multiplies those values together to get the sum.

Github Link:

<https://github.com/Jack-Braun/Jack-Braun.github.io.git>

References / Content Declaration:

main banner picture, by Charlotte Noelle:

<https://unsplash.com/photos/98WPMlTl5xo>

photo of Vancouver, photographed by Aditya Chinchure:

<https://unsplash.com/photos/2yN10KHNTzM>

France travel video:

<https://www.youtube.com/watch?v=e2wm_VRROvA>

USA travel video:

<https://www.youtube.com/embed/biC7sU89rTE>

Canada travel video:

<https://www.youtube.com/embed/V8s9EkYsTSc>

Reviewer photograph, by Foto Sushi:

<https://unsplash.com/photos/6anudmpILw4>

dog photo, by charlesdeluvio:

<https://unsplash.com/photos/Mv9hjnEUHR4>

I was having some trouble with the range input, so I used this to help me learn:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range>