CSCU9B2 – Assignment Report

Structure and JavaScript

My website is made up of seven files: index\_2716761, order.html, contact.html, about.html, 2716761Style.css, shapes.js and CSCU9B2ProjectLogoGrey.JPG.

Each HTML page is made up of the logo (the JPG) at the top left which links back to index\_2716761.html, a navbar at the top which links to the order, about and contact pages and use the 2716761Style.css stylesheet.

Most of the content lies in the order.html page. It is composed of five parts.

1. The script in the header that sets up the page based on the basic cost and the functions that deal with bag orders.
2. A script that checks if the user is using Internet Explorer and if so, warns the user to use another browser.
3. The canvas on which the bag, the text and the colour buttons are drawn.
4. The JavaScript file shapes.js where the bag drawing functions are and where the information about the canvas is stored.
5. The text field and buttons that allow to specify what text to put on the bag, how many bags to order and to place the order.

First of all, the scripts on the order page fetch the cost per bag (of 3p) from the bottom of the html file. A variable individualCost is given the base cost and then calls setBasicCost(individualCost) which then displays the base cost and calls calculateCost(100) which updates the provisional total display to the price of 100 bags. The next functions add100() and subtract100() are called when the corresponding buttons are clicked. They fetch the current number of bags from the display, add or subtract one hundred and then call calculateCost() with the new number of bags. The final function in that script is placeOrder() which simulates the placement of an order with a pop-up window that tells the user how many bags he ordered, how much the order is worth and all the information about the bag through getBagStats() which returns bag colour, text colour and what the text is. This last function is in shapes.js because that is where the bag information is stored.

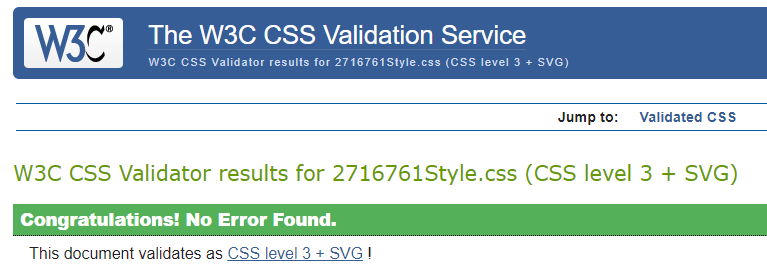
The canvas is given the proper size to contain the bag and the colour buttons, as well as a border to set it apart from the rest of the page, to show that it’s preview of what the bag will be. The buttons call functions which change the colour or textColour variables in shapes.js. The new colour is hard coded in the buttons and the buttons text also contains the name of the colour, to be accessible to screen readers (the underscores after the colour names and the monospace font make all texts the same lengths, forcing buttons to be the same size).

The shapes.js file sets up the canvas, first storing the canvas, getting the context, setting the bag, text and background colours as well as the coordinates for the bag to be drawn and the text to be shown on it. The next three functions create shapes that are used to draw the bag: a line, a quadrilateral and an ellipse. They are all used in drawBag(bagX, bagY, colour) which draws the bag of a specific colour onto the canvas. The drawCanvas() function draws the bag as well as the texts that prompts the user to click the colour changing buttons and the text onto the bag. The next functions are the colour changing functions and submitForm(). It fetches the data from the input, sets the text variable to that text, empties the input, puts the focus to that input and then redraws the canvas. The last function is getBagStats() which compares the hex codes of the colours and converts them to their names and returns a sentence containing them and the text. Finally, drawCanvas() is called so that the canvas doesn’t appear blank.

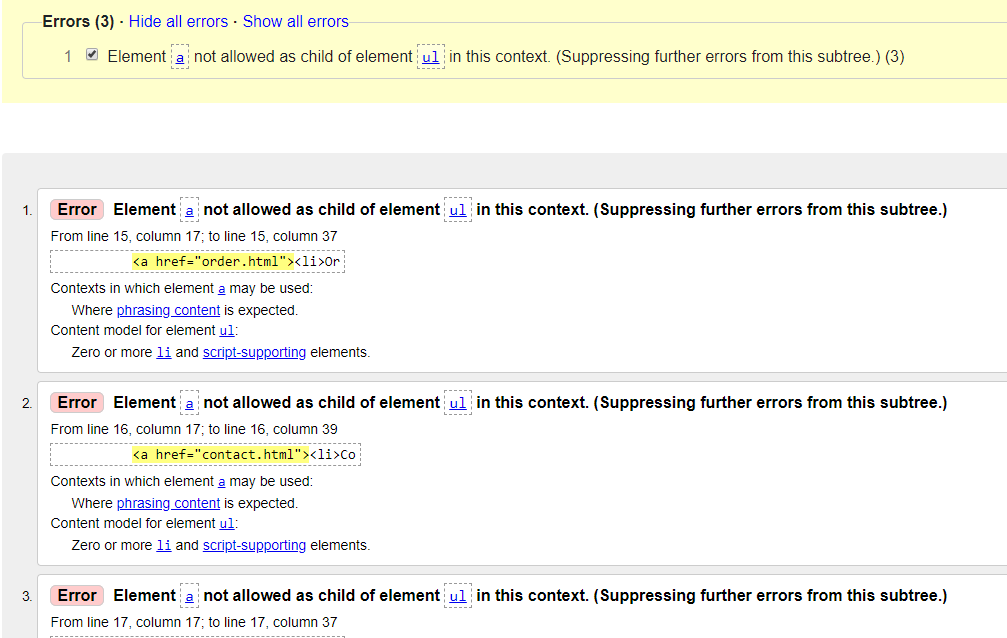
The final part of the order page is the form that asks the user for text to be put on the bag and a submit button that calls submitForm(), the adding and subtracting buttons as well as a small text field that contains the current number of bags to be ordered, the text fields where the costs will be displayed and finally the place order button.

The other web pages have some content that you would typically find on a company’s website, like information about the firm and a way to contact them. The e-mail form on contact.html doesn’t work because in order to stay anonymous I cannot setup my email address to be the recipient.

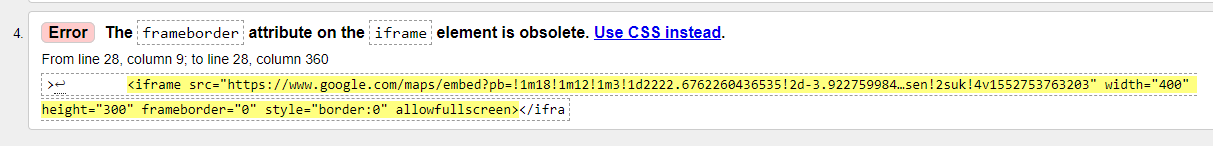
Code Validation



Validation of the CSS, no errors.



All four webpages have the errors above, as they each use the same navbar. I chose to put the <a> tag before the <li> tag to make the whole of the cell clickable and not just the text.



These errors are in the index\_2716761.html file and are caused by the interactive map, the code for which is generated by google maps and the code runs regardless.