Web Application Development Project Submission 2022

Name: Sarah McNelis

Student ID: G00398343

Date: 16/05/2022

Home page: "homepage.html"

Project Requirements Implementation

ITEM 1	Reference
Allow the customer to enter their login details and have login details validated.	See page "login.html" – lines 112 – 130.
Brief description of how this was implemented:	▪ This was implemented using a form action attribute. Once email and password are validated, the action attribute in the form tag on line 117 of "login.html" allows the user to access the "recipes.html" page. ▪ Within this form tag I have used input tags on lines 120 and 124 from "login.html". This allows the user to input the required information. ▪ Next, I have used a button tag on line 127 of "login.html" which allows the user to submit their information. If details are valid, then the action attribute already mentioned will re-direct the user to the recipe page.

ITEM 2	Reference
Perform form validation through JavaScript or HTML to ensure that text fields are not empty, and a valid email address is entered.	See “login.html” lines 120 and 124.
Brief description of how this was implemented:	▪ I used HTML5 validation for email and password. ▪ For the email address on line 120 I have added a type attribute to equal email. This allows HTML to ensure a valid email address is entered. It will not allow the page to redirect until a valid one is entered. ▪ Similarly, on line 124 I have used the type attribute again within the input tag to set the type is password. This will mask the user’s password for security purposes. ▪ Finally, for both input tags I have included the required attribute. This will refuse redirection if either field is empty. Once both fields have been filled and validated the “login.html” redirects the user to “recipes.html”.

ITEM 3	Reference
Access and change HTML on the web page through the DOM.	A) See “homepage.html” – lines 115-119 and lines 140-143. B) See “gallery.html” – line 141 and lines 271-274.
Brief description of how this was implemented:	A) On line 116 of “homepage.html” the user can enter their name and click the button on line 117. This will run the JS function found on line 140. This function uses DOM to retrieve the value of the user’s input and then create an alert/pop-up welcoming that name to the page. This is done using the DOM query: document.getElementById("name").value; B) On line 141 of “gallery.html” I have given an id for this heading which is used in line 273. This creates an alert of the HTML h1 tag using DOM. This is done using the DOM query mentioned in example A again : document.getElementById("popMessage").innerHTML; This will read the inner HTML tag relating to that id and re-produce it on the alert.

ITEM 4	Reference
Access and change styling through the DOM.	A) See “about_mammy.html” – line 85 and lines 148-152. B) See “about_mammy.html” – line 133 and lines 158-164. C) See “gallery.html” – line 139 and lines 277-281.
Brief description of how this was implemented:	A) On line 85 of “about_mammy.html” I have used another event button which will trigger the function changePara() on line 158. This function includes changing that id’s paragraph background color, border details, font, and a box shadow. This is all done was using the DOM query: document.getElementById("new Pará"); and then adding .style followed by the HTML objects that I wish to change. B) On line 133 of “about_mammy.html” I have added another button on the bottom of the page. This will run the changeBackground function on line 150. In this function I have used the .style function directly with my DOM query instead of creating a variable like I did in example A. This will change the background color of the entire page once the button is clicked. C) Again, on line 139 of “gallery.html” I have used another button to change the background color. Once clicked, the changeBackground function on line 279 will be run and the background color will change.

ITEM 5	Reference
Demonstrate the use of events.	I have used events throughout my website. A) See “homepage.html” line 117. B) See “about_mammy.html” – lines 85 and 133. C) See “gallery.html” – line 139. D) See “reviews.html” – lines 163 and 219.
Brief description of how this was implemented:	All of these examples use the event “onclick”. This means that once the button is clicked, the onclick value is triggered. In all of these cases a JS function is activated. These functions allow for a number of tasks to be performed. For example: A) Event on “homepage.html” line 117 allows for function on line 140 to be generated producing an alert. B) Event on line 85 of “about_mammy.html” triggers the function on line 158 to change the paragraph style using DOM. Also, the event on line 133 starts the function on line 150 changing the background color. C) Similarly on line 139 of “gallery.html” the event allows the background color to change using the function on line 279. D) Finally, the events used in “reviews.html” both run functions which create charts from csv and an array when the buttons are clicked.

ITEM 6	Reference
Contain two D3 data visualisations (e.g., Bar Chart) of your choosing. ***a. One from a CSV file*** ***b. One from an array***	See “reviews.html” A) CSV D3 Visualization - see HTML div from line 114-169 and JS function from lines 247-424. B) Array D3 Visualization – see HTML div from line 172-225 and JS function from lines 427-591.
Brief description of how this was implemented:	To access “reviews.html” you must connect to http server within my project folder. A) The function to create a chart form csv data involved the following steps: ▪ The csv file was read in using D3. This csv can be found within my project folder and is named “cousins_review.csv”. Console.log functionality allows for the data to be read in and used. ▪ Next, I created variables. This was done as most of these variables are used multiple times in later code. Therefore, if I ever wanted to change some of these variables in the future. I only need to do so in from lines 257-264. ▪ Following that, I needed to convert the csv array into numbers as they were contained within the csv array as strings. This allowed me to access the rating scores. ▪ Next step was to create a scale for Y and X. ▪ After that I created axis for both Y and X. ▪ Then, on line 305, I used the d3.select method to remove any data created in the chart so that if I create another one it will replace that chart instead of duplicating it.

- Next, I created the SVG container and append attributes at line 309.
- Then I created the rectangle at line 316. And I passed in the data from the CSV here.
- On line 322, I entered attributes to the rectangle.
- However, I have used animations like transitions for both the chart and labels. This starts on line 324 where I added start state attributes for color, x, y, height, and width.
- Next starts the actual transition followed by the end state attributes on line 341.
- Then on line 359 I have added a label and am appending attributes.
- Again, the labels have attributes for the start of the transition, then the transition happens and then attributes are assigned for the end state of the transition.
- Finally, the last steps was to create an axis for x and y and add text. This was done by appending to the group of the SVG container.

B) The function to create a chart from an array starts on line 427. It mostly takes the same format and example A. However, there are a few differences.

- One of the main differences is that instead of reading in a csv I have created arrays with dummy data and values. This data is given variable names which will be used later in the code.
- Next, same as option A, on line 438 I created variables for data which will be used multiple

	times throughout this function. - From line 448 onwards. I have used the exact same format as example A. The only difference is that I am using data from an array via a variable name instead of reading it in from a csv file.

ITEM 7	Reference
Both visualisations should allow the user to specify display settings, including an option to change colour, display size and animations.	See "reviews.html" A) CSV D3 Visualization - see HTML div from line 114-169 B) Array D3 Visualization – see HTML div from line 172-225
Brief description of how this was implemented:	A) Another reason I used variables in my function code was so I could use DOM to retrieve user choices and use those choices for some attributes in my function. - From line 132 - 138 I have used a select tag with options tags inside. These allow the user to choose the height of the chart. The chosen value is then passed into the function using DOM on line 257. - From line 142 – 147 is another block of options tags within a select tag. Again, the user can choose the chart width. This is then passed into the function using DOM at line258. - Next, the user can choose the duration of the transition in the select/options tags block from line 151-156. Again, this is passed into the variable on line 260.

	▪ Finally, the user can choose a color for the chart using the color picker tool from the type attribute. This is passed into the variable on line 259. ▪ Once the button is clicked all this information is retrieved by DOM queries in the function from lines 247-424. B) Again, example B has the very same steps as example A except that the attributes are added to the function where data is read from an array. ▪ From line 189-194 the user can choose the chart height using the options dropdown. This is passed into a variable on line 438 of the function. ▪ From line 198-203 the user can pick the chart width from the options list. This is then passed into the function using DOM at line 439. ▪ The user can then choose the duration of the transition from the options block from line 207-212. Again, this is passed into the variable on line 441. ▪ Finally, the user can choose the color of the chart. This is passed into the variable on line 440. ▪ Once the button is clicked all this information is retrieved by DOM queries in the function from lines 438-441.

ITEM 8	Reference
Have a minimum of 3 linked pages;	See: 1) “homepage.html” – lines 74-91 and line 126. 2) “about_mammy.html” – lines 58-75 and line 138. 3) “gallery.html” – lines 117-134 and line 261. 4) “login.html” – lines 75-92. 5) “recipes.html” – lines 46-63. 6) “reviews.html” – lines 81-98
Brief description of how this was implemented:	All of the above pages are interlinked using the center tags at the top of all of my html files. This tag allowed me to create a header for each tab of my website. 1) “homepage.html” contains the center tag with header links along with a link on line 126 which if clicked will re-direct the user to “reviews.html”. 2) “about_mammy.html” contains the header with links in the center tag and also a link on line 138 which if clicked brings the user to “gallery.html” 3) “gallery.html” again contains the center tag but also a link on line 261 which brings to user to “login.html” in order to access recipes. 4) “recipes.html” only contains the links in the center tag. 5) “reviews.html” also only contains the links in the center tag. All the center tags contain 5 “a” tags which defines the hyperlinks to other pages. Within these tags is an image for each page. Once the image is clicked the hyperlink re-directs the user to the chosen page.

	It is important to note that the center tags only contain 5 out of 6 of my html pages. This is because the "recipes.html" page can only be accessed through the "login.html" page but only when a valid email and password is entered.

Additional information:

All of my code was researched, located, and adapted from online sources. Along with these resources, my code was heavily influenced by my Web Application Development module lectures and notes. This module was the foundation for my website development. My references have been added to my html files using comments.