# Web Application Development Project Submission 2022

Name: **Katie Mc Donald**

Student ID: **G00398279**

Date: 12**th May 2022**

Home page/index page/start page (eg., page user should open first): **index.html**

# Project Requirements Implementation

|  |  |
| --- | --- |
| **ITEM 1** | **Reference** |
| *Allow the customer to enter their login details:* | See page “login.html” |
| *Login details validated (via a login screen)* | The form was created using HTML and styled using internal CSS. |
| *Username:* | python@planted.ie |
| *Password:* | python3 |
| *Brief description of how this was implemented:* | A main container “container” was created using HTML. Inside “container” is a sub-container “main-holder” is created. In “main-holder” under the h1 login title there is a container “login-error-msg-holder” which holds a “login-error-msg” that appears when a user incorrectly enters an email and/or password.  To create the “login-form”, the <form></form> tag is wrapped around the input fields for “Email” and “Password”, along with their labels and a “login-form-submit” button using HTML.  The styling for all elements on the page (apart from the navigation bar) is done through internal CSS. The main container and sub containers are centred on the page. The opacity of the “login-error-msg” is set to 0, so it is hidden when the page is first loaded. |

|  |  |
| --- | --- |
| **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* | Validation is preformed through HTML5 and JavaScript |
| *Brief description of how this was implemented:* | HTML5 validation has been added to the input fields by marking them “required”. Additionally, the “Email” input has a regex pattern which ensures an email address format is adhered to [1].  The HTML DOM method getElementById is used to get variables.  The authentication of details is performed by the JavaScript function “log” found on the page between the <script></script> tags. The function returns the value properties entered into the “Email” and “Password” fields and using comparison operators checks them against the values [python@planted.ie](mailto:python@planted.ie) and “python3”. The function also uses conditional statements in the event either field is left blank or incorrect details are entered. In the latter, the opacity of “login-error-msg” changes for it to become visible to the user. An alert confirms to a user that they have successfully logged in. |

|  |  |
| --- | --- |
| **ITEM 3** | **Reference** |
| Access and change HTML on the web page through the DOM; | See page “dom.html”.  The “dom.html” page is split into three thirds – left-half, center, and right-half. |
| *Brief description of how this was implemented:* | In “left-half” the JavaScript function “text” completes a Mad Lib by retrieving the user input value to “input1” and using the HTML DOM method getElementById access the HTML and then manipulate the Mad Lib by appending the retrieved “input1” value by innerHTML.  In “center” there is a button with the id “datetime” which when clicked the JavaScript function “clicked” uses a DOM method to access the HTML and uses innerHTML to append a user’s browsers time zone and date to the h3 heading with the “tine” id.  Also, in “center” the HTML is changed by the JavaScript function “rubber”. The function activates when the button with the id “del” is clicked. Using a DOM method, the HTML text with the id “p2” is accessed and deleted from the page using the DOM remove method.  In “right-half” when the HTML text with the id “ouch” is clicked, the HTML is access and changed by the JavaScript “changeText” function. The function works by access the visible text by a DOM method, getElementById and uses innerHTML to change the text on the page.  Finally, in “right-half” a JavaScript function “myName” uses a DOM method to retrieve the user input and append the “input2” value to a sentence using innerHTML. The resulting sentence is shown via an alert to the user. |

|  |  |
| --- | --- |
| **ITEM 4** | **Reference** |
| Access and change styling through the DOM; | See page “dom.html”.  The dom.html page is split into three thirds – left-half, center and right-half. |
| *Brief description of how this was implemented:* | In “left-half” there is a paragraph explaining DOM. The paragraph is within a div with the id tag “output-text”. The font-style of this paragraph can be changed by a dropdown menu, allowing the user to select a different font-style, this is performed by the JavaScript function “changeFontStyle” found on the page between the <script></script>. When the default font-style is changed the function uses a HTML DOM method getElementById to update the font-style.  Additionally, the font-colour of the paragraph can be changed using DOM through selecting a colour via a colour picker. The default font-colour is black. A JavaScript event notes when the font-colour is no longer black, when this happens the function “colourChange” found between <script></script> tags, uses a DOM query to extract the newly selected colour, which in turn activates the JavaScript function “updateColour”. This function uses a DOM query selector to apply the updated the font-colour to the text between the <p></p> tags.  The background colour of the page can also be changed through DOM. In “center” there is a button with the id “bg”, when clicked the JavaScript function “change” found on the page between the <script></script> tags, changes the background colour of the page. It performs this by iterating through an array of colours and uses a DOM query selector to change the colour of the page background to a colour within the array and applying it to the <body> tag. |

|  |  |
| --- | --- |
| **ITEM 5** | **Reference** |
| Demonstrate the use of events; | See page “dom.html”.  The dom.html page is split into three thirds – left-half, center and right-half.  Events can also be seen on – “login.html”, “stats.html”, “gallery.html” and “contact.html”, all have “onclick” events. |
| *Brief description of how this was implemented:* | In the “left-half” the font-colour of the paragraph explaining DOM can be changed. A JavaScript event found between the <script></script> tags is used to get the default value of the colour picker and the new user selected colour through a handler function and applies the newly chosen font-colour to the to the text between the <p></p> tags.  In “center” there is a drag and drop event. The drag event occurs when the user selects the text with the id “dragtarget” and drags the element into the “droptarget” element and releases the “dragtarget”.  An event is used to change the CSS styling of the “droptarget” when the “dragtarget” has been released into it.  Also, upon its release into the “droptarget” an event is used to append text to the “droptarget” box. This gives the appearance of the text sitting inside the target.  In the “right-half” an onmousemove event activates the JavaScript function “mouseCount”. The function keeps score of the number of times the mouse is moved within the “mouse-over” div. The function uses the HTML DOM method getElementById to update the count using innerHTML.  Additionally, events are seen throughout the website – “login.html”, “stats.html”, “gallery.html” and “contact.html” all have buttons with “onclick” events. “dom.html” contains additional “onclick” events which have already been mentioned above.  In “gallery.html” an “onclick” event allows the user to click on “next” and “prev” arrows allowing a user to move though a slideshow. An “onclick” is also applied to the dots which correspond to an image, providing another way to move through the slideshow.  In “contact.html” an “onchange” event is added to the <input> tag for the “date”. When the date is changed the event activates the JavaScript function “dateCheck” which use DOM HTML to retrieve the users chosen value and using a comparison operator compares it to the current day, ensuring that the user selected value is valid (i.e., cannot be a day in the past). |

|  |  |
| --- | --- |
| **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 page “stats.html”.  The stats page is divided in half. The “left-half” contains the D3 visualisation from a csv file and the “right-half” contains the D3 visualisation from an array.  The csv file is in the main project folder named “bookings.csv” and the array is found between the <script></script> tags and is clearly marked. The values used for the array have been taken from <https://financesonline.com/restaurant-statistics/>. |
| *Brief description of how this was implemented:* | Both D3 visualisations are created using a JavaScript function; “drawChart” for “left-half” and “drawChart2” for “right-half”. They are found between the <script></script> tags. In the “left-half” a <div> with the id “myDiv” will show the D3 chart generated by the JavaScript “drawChart” and the <div> with the id “svg\_div” displays the generated D3 chart from the JavaScript function “drawChart2”. Both functions are activated when the user clicks the buttons with the id’s “draw” and “draw2” respectfully are clicked.  To retrieve the data from the csv file, the “drawChart” function loads the data using the D3 method csv and use a function to print the data to the console. The data must be converted to numeric values as they are currently in a string format. For the csv chart to fully work please run the site with a python web server. See below in the “Additional information” section for further details on how to run the server.  For both the csv and array charts a yScale is created using the D3 “scaleLinear” method. This creates a “visual scale point” [2] by taken a value from the domain (“Bookings” for the csv and “value” for the array) and returns the corresponding values from their range.  The xScale is created by using the D3 “scaleBand” method. This method will “…divide the range evenly between the elements of the domain” [3]. The scale is applied to the domain (“Month” for the csv and “key” for the array) and returns the start of the band.  The x and y axes are created using the newly created scales.  To prevent multiple charts being created within the output <div>’s we use the select and remove methods to select all content within the area and delete it.  Next a SVG container is created and appended to the output <div>. To this SVG container we create a rectangle whose values relate to either the csv or array data. Attributes are added to the rectangle such as positioning on the axes, gap between bars and any animation to the bar chart bars.  Labels are then created and appended to the SVG container whose values relate to either the csv or array data. Attributed can be added to the labels such as their positioning, font-size, animation etc.  The final part of both the “drawChart” and “drawChart2” JavaScript functions is adding the x and y axes to the SVG containers. This must always be done last, so it is visible by sitting on top of the bars. |

|  |  |
| --- | --- |
| **ITEM 7** | **Reference** |
| Both visualisations should allow the user to specify display settings, including an option to change colour, display size and animations | See page “stats.html”.  The stats page is divided in half. The “left-half” contains the D3 visualisation from a csv file and the “right-half” contains the D3 visualisation from an array. |
| *Brief description of how this was implemented:* | In both “left-half” and “right-half” there is a dropdown menu, which allows a user to select the height of the D3 visualisation. Between the <script></script> tags, the selected height value is extracted by the DOM HTML method getElementById and stored in the “height” variable. In their respective JavaScript functions, “drawChart” for “left-half” and “drawChart2” for “right-half” the “height” variable is used in place of hard coded values. This allows the user to determine the size of the chart.  Again, in both “left-half” and “right-half” there is a colour picker, which allows the user to select the colour of the bars for the D3 visualisation. Between the <script></script> tags, the selected colour is extracted by the DOM HTML method getElementById and stored in the “chartColour” variable. In their respective JavaScript functions, “drawChart” for “left-half” and “drawChart2” for “right-half” the “colourChart” variable is used in place of a hard coded values. This allows the user to determine the final colour of the D3 chart.  Additionally, another dropdown menu is included in both “left-half” and “right-half” allowing the user to select the speed duration of the D3 visualisation animation. Again, between the <script></script> tags, the selected speed value is extracted by the DOM HTML method getElementById and stored in the “speed” variable. In their respective JavaScript functions, “drawChart” for “left-half” and “drawChart2” for “right-half” the “speed” variable is used in place of hard coded values. This allows the user to determine the speed of the D3 chart animation. |

|  |  |
| --- | --- |
| **ITEM 8** | **Reference** |
| Have a minimum of 3 linked pages; | The website is a business page for a restaurant called “Planted”. It consists of 8 linked pages: “index.html”, “about.html”, “menu.html”, “gallery.html”, “dom.html”, “stats.html”, “contact.html” and “login.html”. |
| *Brief description of how this was implemented:* | Each webpage is linked by HTML5. A navigation bar at the top of each page contains a HTML <a> “href” attribute which specifies a relative URL that redirects a user to another page within the website.  The navigation bar is found within the <header></header> tags on each page. The navigation bar is created by placing the link between unorder list tags <ul> and further wrapping each <a> “href” link within a listed item <li> tag.  The CSS properties for the navigation bar is done via external CSS. The style sheet named “style.css” can be found in the main project folder. It is linked to each page via <link> tag which is wrapped by the <header></header> tags and contains a “href” attribute to the CSS file. I chose to use external CSS for the navigation bar because it was the one element which was include on each page. |

## Additional information:

The website should be ran while using the Python web server in the same folder to facilitate D3.js elements and to maximize interactive experience. The csv D3 data visualisation will not work otherwise. To run the python webserver:

1. Navigate to the project folder location through the command prompt.
2. Enter "Python -m http.server" on the command line.
3. Let serve run in the background Leave the server running in the background.
4. Open browser and enter <http://localhost:8000/> in the search bar.
5. Open the site page.

The Instagram and Facebook social media icons on seen on “index.html” do not link to another page.

The “bookings.csv” file found in the project folder contains the data used for the D3 data visualisation in the “left-half” of the stats.html page.

There is also a images folder within the main project folder which contains all images used for the website.

A google map to show the location of the restaurant was generated using map.ie’s “Create a Map” option. This generates a responsive Google Maps HTML iframe which can be added to a website.

# References

|  |  |
| --- | --- |
| [1] | How can I validate an email address in JavaScript? [Internet]. Stack Overflow. [cited 2022 April 17]. Available from: https://stackoverflow.com/questions/46155/how-can-i-validate-an-email-address-in-javascript |
|  |  |
| [2] | D3.js scaleLinear() method [Internet]. GeeksforGeeks. 2020 [cited 2022 May 1]. Available from: https://www.geeksforgeeks.org/d3-js-scalelinear-method/ |
|  |  |
| [3] | Fil. d3.scaleBand [Internet]. Observable. 2019 [cited 2022 May 1]. Available from: https://observablehq.com/@d3/d3-scaleband |
|  |  |
| [4] | Duignan M. Lecture notes for Web Application Development, Higher Diploma in Science (Data Analytics), Galway Mayo Institute of Technology. 2021. |
|  |  |
| [5] | SVG being cut off [Internet]. Stack Overflow. [cited 2022 May 4]. Available from: https://stackoverflow.com/questions/24894921/svg-being-cut-off |
|  |  |
| [6] | Anthony J. 85 significant restaurant statistics: 2021 analysis of data & market share [Internet]. FinancesOnline.com. 2019 [cited 2022 May 1]. Available from: https://financesonline.com/restaurant-statistics/ |
|  |  |
| [7] | Set space between divs [Internet]. Stack Overflow. [cited 2022 Apr 30]. Available from: https://stackoverflow.com/questions/10052922/set-space-between-divs |
|  |  |
| [8] | Kumar N. How the question Mark (?) operator works in JavaScript [Internet]. freeCodeCamp.org. 2021 [cited 2022 May 2]. Available from: https://www.freecodecamp.org/news/how-the-question-mark-works-in-javascript/ |
|  |  |
| [9] | How to change font style using drop-down list in JavaScript ? [Internet]. GeeksforGeeks. 2020 [cited 2022 May 2]. Available from: https://www.geeksforgeeks.org/how-to-change-font-style-using-drop-down-list-in-javascript/ |
|  |  |
| [10] | JavaScript HTML DOM [Internet]. W3schools.com. [cited 2022 April 27]. Available from: https://www.w3schools.com/js/js\_htmldom.asp |
|  |  |
| [11] | HTML onmousemove event attribute [Internet]. W3schools.com. [cited 2022 April 28]. Available from: https://www.w3schools.com/tags/ev\_onmousemove.asp |
|  |  |
| [12] | Handling events in JavaScript [Internet]. JavaScript Tutorial. 2020 [cited 2022 April 28]. Available from: https://www.javascripttutorial.net/javascript-dom/handling-events-in-javascript |
|  |  |
| [13] | JavaScript date objects [Internet]. W3schools.com. [cited 2022 May 1]. Available from: https://www.w3schools.com/js/js\_dates.asp |
|  |  |
| [14] | JavaScript DOM HTML [Internet]. W3schools.com. [cited 2022 April 25]. Available from: https://www.w3schools.com/js/js\_htmldom\_html.asp |
|  |  |
| [15] | Maps.ie. [cited 2022 May 7]. Available from: https://www.maps.ie/create-google-map/ |
|  |  |
| [16] | Pixel PX to VH [Internet]. Github.io. [cited 2022 April 30]. Available from: https://khaledkzy.github.io/pixel-vh-vw-converter/ |
|  |  |
| [17] | HTML color picker [Internet]. W3schools.com. [cited 2022 April 25]. Available from: https://www.w3schools.com/colors/colors\_picker.asp |
|  |  |
| [18] | How to make a Navigation Bar in HTML [Internet]. www.javatpoint.com. [cited 2022 April 21]. Available from: https://www.javatpoint.com/how-to-make-a-navigation-bar-in-html |
|  |  |
| [19] | How to create social media buttons [Internet]. W3schools.com. [cited 2022 May 12]. Available from: https://www.w3schools.com/howto/howto\_css\_social\_media\_buttons.asp |
|  |  |
| [20] | |  | | --- | | CSS z-index property [Internet]. W3schools.com. [cited 2022 May 3]. Available from: https://www.w3schools.com/cssref/pr\_pos\_z-index.asp | |
|  |  |
|  |  |

## Images

* https://www.supaldesai.com/blog/2019-03-03-3-epic-meals-you-cant-miss-in-london
* https://static.vecteezy.com/system/resources/previews/002/946/324/large\_2x/monstera-plant-leaves-isolated-on-pink-background-free-photo.jpg
* <https://media-cdn.tripadvisor.com/media/photo-s/21/0d/a5/9c/creamy-pumpkin-coconut.jpg>
* https://resizer.otstatic.com/v2/photos/xlarge/1/26638508.jpg
* <https://zestfulkitchen.com/wp-content/uploads/2019/12/sweet-potato-ravioli_cover-673x740.jpg>
* <https://veganheaven.org/wp-content/uploads/2018/06/Vegan-Dessert-Recipes-7-683x1024.jpg>
* <https://media-cdn.tripadvisor.com/media/photo-s/16/de/3a/f4/mushroom-satay-ramen.jpg>
* <https://i.pinimg.com/originals/cd/50/06/cd5006b8e734619171afe8502acf7917.jpg>
* <http://veginity.com/images/Whats_on/Vietnamese/Vietnamese_Composite_2.jpg>
* <https://i.pinimg.com/originals/b2/18/c3/b218c3f9c578b841ccbe59a8117cbbc6.jpg>
* <https://vegnews.com/media/W1siZiIsIjE3NDE0L1dvbHQtMTguMDguMTAtUGxhbnQtUG93ZXItRm9vZDI4NjEtc2NhbGVkLWUxNTc3NDQxODIwOTg5LmpwZyJdLFsicCIsImNyb3BfcmVzaXplZCIsIjEwMDB4NTkxKzArMzciLCIxNjAweDk0Nl4iLHsiZm9ybWF0IjoianBnIn1dLFsicCIsIm9wdGltaXplIl1d/Wolt-18.08.10-Plant-Power-Food2861-scaled-e1577441820989.jpg?sha=9345f3bec1dad6ad>
* <https://static.visitestonia.com/images/3714880/600_400_false_false_8714384badff3ba55f53fc542d460357.jpg>
* <https://biancazapatka.com/wp-content/uploads/2019/04/mushroom-pasta-spinach-best-recipe-easy-vegan-healthy-pasta-spinat-rezept-einfach-champignons-tasty.jpg>
* <https://www.honeywhatscooking.com/wp-content/uploads/2021/05/B8A9DFCE-A8E5-401F-8FD5-DFA9896B7752_1_105_c.jpg>
* <https://quitegoodfood.co.nz/mushroom-risotto-saffron-sage/>
* <https://www.tripadvisor.com/LocationPhotoDirectLink-g274958-d6612483-i265850611-Vegan_Restoran_V-Tallinn_Harju_County.html>
* <https://www.veganlifemag.com/wp-content/uploads/2018/05/MushroomPho.jpg>
* <https://www.abelandcole.co.uk/media/1946_16335_x.jpg>
* <https://media-cdn.tripadvisor.com/media/photo-s/14/f4/ea/e5/portobello-mushroom-pate.jpg>
* <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQgvKy1p5wxXDsTvGV3u5_poJKXuSw0TB6x4g&usqp=CAU>
* <https://addictedtodates.com/vegan-oreo-cheesecake/>
* <https://data.thefeedfeed.com/static/2020/01/27/15560504305cbf71fe6f09a.jpg>
* <https://data.thefeedfeed.com/recommended/post_4491709.jpg>