

Website Implementation Report -Food Cart Brisbane

Tahseen islam

Student ID: 45808609

2022

Table of Contents

[Table of Figures 2](#_Toc104337840)

[Summary of implementation of HTML, CSS, and JS 1](#_Toc104337841)

[HTML 1](#_Toc104337842)

[CSS 1](#_Toc104337843)

[JS 1](#_Toc104337844)

[Screenshots of website 2](#_Toc104337845)

[Home page 2](#_Toc104337846)

[All Food Page 3](#_Toc104337847)

[Detailed Food Page 3](#_Toc104337848)

[FAQ Page 4](#_Toc104337849)

[Contact Page 4](#_Toc104337850)

[Challenges faced and lessons learned 5](#_Toc104337851)

[Nielsen’s Usability Heuristics 5](#_Toc104337852)

[#1: Visibility of system status 5](#_Toc104337853)

[#2: Match between system and the real world 5](#_Toc104337854)

[#3: User control and freedom 5](#_Toc104337855)

[#4: Consistency and standards 5](#_Toc104337856)

[#5: Error prevention 6](#_Toc104337857)

[#6: Recognition rather than recall 6](#_Toc104337858)

[#7: Flexibility and efficiency of use 6](#_Toc104337859)

[#8: Aesthetic and minimalist design 6](#_Toc104337860)

[#9: Help users recognize, diagnose, and recover from errors 6](#_Toc104337861)

[#10: Help and documentation 6](#_Toc104337862)

[Conclusions and future work 7](#_Toc104337863)

[References 7](#_Toc104337864)

# Table of Figures

[Figure 1: Home page desktop view 2](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337865)

[Figure 2: Home page mobile view 2](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337866)

[Figure 3: All food page mobile view 3](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337867)

[Figure 4: All food page desktop view 3](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337868)

[Figure 5: Detailed food page desktop view 3](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337869)

[Figure 6: Detailed food page mobile view 3](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337870)

[Figure 7: FAQ page desktop view 4](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337871)

[Figure 8: FAQ page mobile view 4](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337872)

[Figure 10: Contact page mobile view 4](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337873)

[Figure 9: Contact page desktop view 4](https://d.docs.live.net/d7deef676712cf77/Semester%201%5eJ%202022/Introduction%20to%20Web%20Design/Islam_Tahseen_Website_Implementation_Report.docx#_Toc104337874)

# Summary of implementation of HTML, CSS, and JS

## HTML

A total of 5 HTML pages have been used to create the Food Cart Brisbane Website which can be viewed [here](https://deco-1400-project.vercel.app/). The first page (index.html) is the landing page which contains a slideshow, and the popular food carts section. The second page (allfood.html) is where all food carts are shown and a search bar, filter and reset button is available to use. Upon clicking any food cart they user will be navigated to the third page, detailed food page (detail.html), where the specific food cart will be shown in detail. The fourth page (faq.html) is the FAQ page and all common questions and answers are provided in the form of an accordion. The fifth page (contact.js) is the contact page where users can fill the form and submit to contact support. The prettier VSCODE extension was used to enforce a strict html format leading to cleaner looking code.

## CSS

A single CSS file (style.css) was used for the styling of all HTML. The CSS is neatly separated into different sections respective of the HTML page. The CSS class naming convention that was used was BEM (Block-Element-Modifier) as it ensures class names are relevant to the section being styled which can later be easily identifiable and modified if necessary. Flexbox was used for the majority of element positioning but position:absolute was used where necessary (e.g. modal). Selectors and pseudo classes such as :hover was used to create the UI for example buttons would have a hover effect and on hover, the background colour would change slightly to notify the user is hovering over the button. On hover nav links would get an underline as well. CSS keyframes was used to create the modal for mobile devices. On opening of the modal, it would have a slide-in effect created using keyframes. Media queries were used to aid in the responsiveness of the website. For example, the contact page is initially set to flex-direction: row but as the screen width gets smaller flex-direction: column is used to fit contents better. Finally using JS toggleable classes were able to be added to show/hide contents for example the accordions uses JS to toggle an ‘active’ class which allows for the accordion answer to be shown when clicked.

Note: A aos.css file exists which is needed for the external plugin AOS (animate on scroll) to be used.

## JS

Various components where created with JavaScript such as the slideshow, modal, displaying of all food trucks, detailed view of food trucks, FAQ accordion, contact form and animating-on-scroll. Pages used getElementByID and querySelector to return the elements to be used for certain tasks. Such as adding an event listener of ‘click’ to be able to run a function when a button is clicked. It was also used to toggle the class list for certain elements so that elements where visible on click e.g. mobile hamburger menu. Asynchronous fetch functions where used to fetch data from a food truck JSON file which was later used to display all food carts (using map array method and innerHTML functions), as well as displaying a single detailed food cart using (find array method, localstorage get and set functions) and filtering the fetched data (via specific user search of an item or a filter button) using the filter and map array methods. An animation plugin was used to create fade in effects on mouse scroll and delayed animations.

Note: A aos.js file exists which is needed for the external plugin AOS (animate on scroll) to be used.

# Screenshots of website

## Graphical user interface Description automatically generatedHome page

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

Figure 1: Home page desktop view

Figure 2: Home page mobile view

## Graphical user interface, website Description automatically generatedGraphical user interface Description automatically generatedAll Food Page

Figure 3: All food page mobile view

Figure 4: All food page desktop view

## A picture containing text, food, screenshot, stew Description automatically generatedDetailed Food Page

Text

Description automatically generated

Figure 5: Detailed food page desktop view

Figure 6: Detailed food page mobile view

## FAQ Page

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 7: FAQ page desktop view

## Contact Page

Graphical user interface, application

Description automatically generated

Figure 8: FAQ page mobile view

Graphical user interface

Description automatically generated

Figure 10: Contact page mobile view

Figure 9: Contact page desktop view

# Challenges faced and lessons learned

A big challenge I faced was trying to figure out a semantic CSS class naming system. What I was coding was getting confusing at some times but after learning BEM for CSS, it made class naming very easy and if there was a styling issue with a certain element in the document, I could quickly find the class using the easy-to-understand class name and edit.

It was also difficult to maintain readable HTML as everything would look cluttered, or I forgot to add some spacing between sections. I learned of a VSCODE extension named ‘prettier’ which is a code formatter that adheres to a certain set of formatting conventions, so everything looked neater.

I couldn’t understand how to read a JSON file in JavaScript and realised I had to use the fetch method to obtain it. I knew I wanted to use this data and use JS to dynamically create the HTML of each food cart detail as manually making a HTML page for each food cart would be tedious and time wasteful. I was able to do this with the array method, map, and dynamically created HTML which would have the carts details.

# Nielsen’s Usability Heuristics

## #1: Visibility of system status

* In the nav bar the current page is underlined so the user knows what page they are on
* In the All-Food Page, the current filter will be highlighted and the current typed search result will be shown so the user doesn’t have to remember / guess

## #2: Match between system and the real world

* Written with simple words for everyone to easily understand
* All information in given in logical order for example the landing page has the nav bar at the top, the main content in the middle and the footer at the bottom.

## #3: User control and freedom

* All buttons and links are self-explanatory and easy to understand
* In the Detailed View Page, there is a back button so users can go back to the previous page
* In the, all food page, users can reset filter and reset search results

## #4: Consistency and standards

* All pages have a similar pattern of nav bar at the top, contents in the middle and footer at the bottom
* All pages main content and buttons are familiar for user e.g. back button means go back to previous page, “search all foods” button takes user to page showing all food carts.
* Contact page follows an easy flow pattern so users can submit info and or directly call/email.
* External links (such as social media links) open in a new tab in case the user wants to continue browsing through current website

## #5: Error prevention

* The contact page error checks the form inputs e.g. the email input will be validated before submit and if it is not valid the form will not be sent and a popup will occur alerting the user of the error.
* When the user is typing a food cart that doesn’t exist, the user will be notified that no food carts match the search result to hint the user to recheck the spelling of the name of a food cart

## #6: Recognition rather than recall

* When searching for food carts, the search result is shown in the input bar. Similarly, when a certain filter is selected, that filter will be shown as the active filter.
* In the contact page, user can see exactly what is being typed and what will be sent when the form is submitted.

## #7: Flexibility and efficiency of use

* Search bar and filter in the all food page is used to aid the user in finding a specific type of food instead of scrolling down.
* A back to top button exists which allows the user to go back to the top of the screen which helps prevent the user from having to manually scroll for a long time to reach the top.
* In the all food page, images are lazy loaded so not all images are not being attempted to download on page load.

## #8: Aesthetic and minimalist design

* The site is simple, easy to understand and navigate.
* Buttons change background colour on hover so users can know it is clickable
* Nav links are underlined on hover so users know it is clickable
* UI is not overloaded with many elements so user is not confused
* Main content is inside a row container which has a max-width of 1300px so if the screen is wider than this, the content will not look stretched, making it hard to read and navigate.
* Site is still functional and is aesthetic even on different devices due to responsive design

## #9: Help users recognize, diagnose, and recover from errors

* Input validation messages popup if invalid or required inputs are not filled in the contact form
* Search results may show a “no results found” if wat the user has typed doesn’t match any food carts

## #10: Help and documentation

* FAQ page provides commonly asked question and the answer to it.

# Conclusions and future work

In conclusion I was able to successfully create a website that is fully functional and am very happy with my work. I have also uploaded it on <https://deco-1400-project.vercel.app/> for it to be viewed online at any time and can even be used to search for food carts in Brisbane.

In the future I will try to add a better way to include the current location of the food cart in Brisbane as the API that the data is called from did not have this information so I will need to figure out how to do that. I will also in the future make the UI look slightly better by adding box shadows, making the website more accessible in terms of contrasting sections and adding animations for the slideshow when going through it. I will also add to the functionality of the filter option as currently it can only filter one food category but if the user wants, they can filter more than one category.

# References

GitHub - michalsnik/aos: Animate on scroll library. (2022). Retrieved 24 May 2022, from https://github.com/michalsnik/aos

Brisbane Food Trucks - Brisbane Food Trucks — Trucks — API - Data | Brisbane City Council. (2022). Retrieved 5 May 2022, from https://www.data.brisbane.qld.gov.au/data/dataset/brisbane-food-trucks/resource/ccd0bd48-5b7f-44bc-b2a2-2891b46f93b3