# Hackathon 2

## Interactive Frontend Development

### Criteria Discussion

# LO1

**Design an interactive Front-End web application using HTML and CSS and JavaScript based on the principles of user experience design, accessibility and responsivity**

# LO1.1 - Design a web application that meets accessibility guidelines, follows the principles of UX design and presents a structured layout and navigation model, and meets its given purpose

Did you check the contrasts between background and foreground?

Have you **alternative text** equivalents for all images?

**Does the user know** where on the site they are?

Does the application meets it's given **purpose**?

# LO1.2 Design a website that meets accessibility guidelines (e.g. contrast between background and foreground colours, non-text elements have planned alternative text equivalents to cater for the visually impaired).

- Give appropriate alternative (alt) text for all images
- Test your contrast ratio
- Your font style should be legible
- Consider appropriate color scheme for the purpose of your site

LO1.3 **Design the organisation of information on the page following the principles of user experience design (headers are used to convey structure, information is easy to find due to being presented and categorised in terms of priority)**

- Ensure user knows the purpose of the site at first glance
- Break the content down into readable parts
- Any interactions expected from the user should be clearly understood and followed
- Ensure that navigating is intuitive

LO1.4 **Ensure that foreground information is never distracted by backgrounds**

- Text should be legible, background color or image should not distract from the content
- Color scheme should not distract or affect the legibility
- Subtle use of CSS animations, appropriate to the need

# LO1.5 **Include graphics that are consistent in style and colour**

- Ensure that images and graphics are consistent in size
- Color scheme is followed with regards to graphics
- There should be overall consistency across the site

## LO1.6 **Design interactivity for a web application that lets the user initiate and control actions, and gives feedback**

- Ensure there is an appropriate feedback on interaction
- Test that the functionalities work as expected
- User should be able to initiate the interaction and receive feedback to their actions
- The feedback should appear at appropriate time and as per design

LO1.7 **Write custom JavaScript, HTML and CSS code to create a responsive front-end web application consisting of one or more HTML pages with significant interactive functionality**

- The site must have at least one HTML page

- It must be responsive on all screens

- The web application must be interactive and produce responses to the user

LO1.8 **Write JavaScript code to produce relevant responses to user actions**

- Include error handling in planning of your application - i.e. if required input is not provided, ensure to include appropriate error handling and feedback to the user

- Use appropriate error handling where user’s input is not as required - feedback to the user should be informative in message, i.e. ‘You did not enter username’, ‘Entry should be a number’ etc

LO1.9 **Implement an interactive web application that incorporates images or graphics of usable resolution, legible, unobscured text, consistent styling, undistracted foregrounds**

- Ensure your images and graphics are of excellent quality, they should not stretch or be pixelated

- Ensure responsiveness on all screens

- Background should not distract from the foreground

# LO2

**Test a front-end web application through the development, implementation and deployment stages**

## LO2.1 **Write JavaScript code that passes through a linter (e.g. Jshint) with no significant issues.**

**Resource:**

- **JS - JSHint**

**Tip:**

- Clear all errors
- If possible, clear all warnings

LO2.2

## Write custom HTML code that passes through the official W3C validator with no issues.

**Resource:**

- **https://jigsaw.w3.org/css-validator**

**Tip:**

- Clear all errors
- Try not leaving any warnings in

LO2.3 **Write custom CSS code that passes through the official (Jigsaw) validator with no issues**

**Resource:**

- **https://validator.w3.org/**

**Tip:**

- Clear all errors
- If possible, clear all warnings

LO2.4

## Code all external links to open in a separate tab when clicked

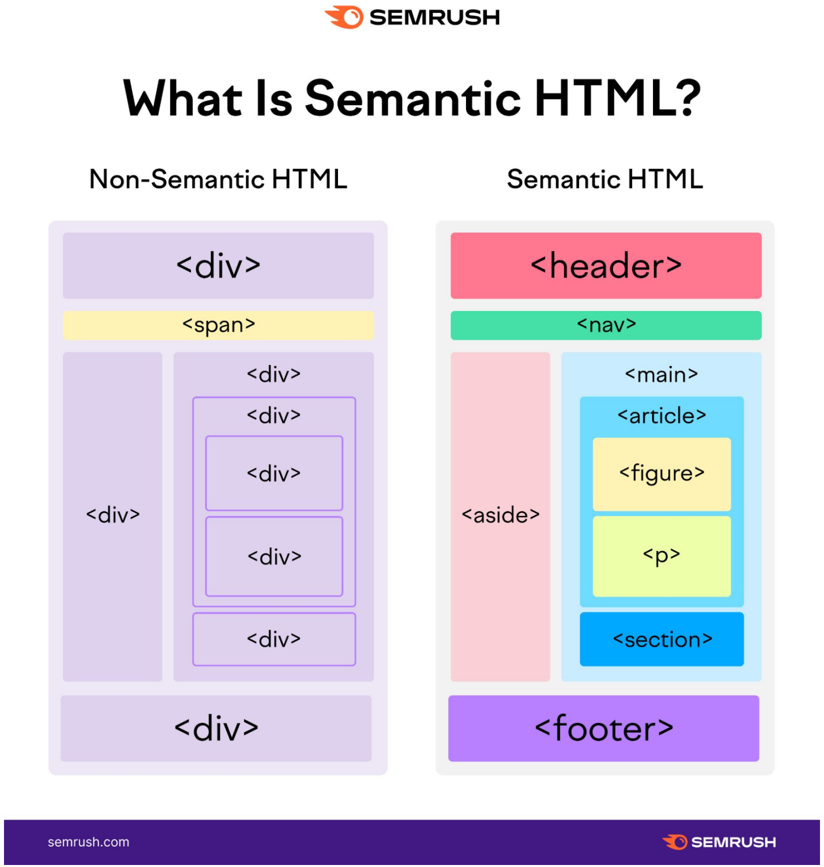
LO2.5 **Use CSS media queries across the application to ensure the layout changes appropriately and maintains the page's structural integrity across device screen sizes.**

- When testing the responsiveness of your site, ensure to check on all screen sizes, use DevTools to see how your site displays across different devices

# LO2.6

**Use Semantic markup to structure HTML code**

- Ensure your site is structured as appropriate



LO2.7 **Present the finished website with clearly understandable site-specific content, rather than Lorem Ipsum placeholder text**

- Even if using fictional information, make it looking as finished as possible
- Do not use Lorem Ipsum
- All content included on the site should be relevant to the project’s purpose as much as possible

## LO2.8 **If used, implement clear navigation to allow users to find resources on the site intuitively.**

Useful resource:

- The Essential Guide to Website Navigation
- Website Navigation Best Practices

# LO3

**Deploy a Front-End web application to a Cloud platform**

LO3.1

## Deploy a final version of the code to a cloud-based hosting platform (e.g. GitHub Pages) and test to ensure it matches the development version

LO3.2 **Use Git & GitHub for version control of an interactive web application up to deployment.**

- Commits small and often
- Message no longer than 50 characters - ensure to address what was changed with this commit

Useful resource:

How to Write a Git Commit Message

LO3.3

# Remove commented-out code before pushing final changes to version control and deploying

LO3.4

# Ensure that there are no broken internal links

# LO4

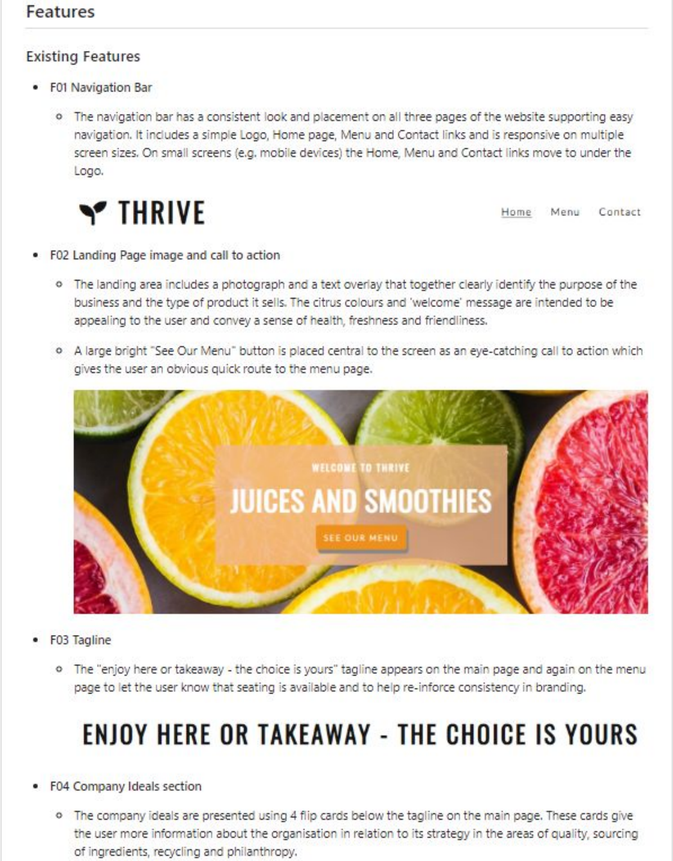
**Maximise future maintainability through documentation, code structure and organisation**

LO4.1 **Write a README.md file for the web application that explains its purpose, the value that it provides to its users, and the deployment procedure.**

- The deployment procedure should be a thorough step by step on how the deployment is performed, so revise how it is done in walkthrough and take down the steps you apply to your own project

LO4.2

# Insert screenshots of the project features, give a brief description of what each feature does and explain its value to the user.



## Features

### Existing Features

- F01 Navigation Bar
  - The navigation bar has a consistent look and placement on all three pages of the website supporting easy navigation. It includes a simple Logo, Home page, Menu and Contact links and is responsive on multiple screen sizes. On small screens (e.g. mobile devices) the Home, Menu and Contact links move to under the Logo.

- F02 Landing Page image and call to action
  - The landing area includes a photograph and a text overlay that together clearly identify the purpose of the business and the type of product it sells. The citrus colours and 'welcome' message are intended to be appealing to the user and convey a sense of health, freshness and friendliness.
  - A large bright "See Our Menu" button is placed central to the screen as an eye-catching call to action which gives the user an obvious quick route to the menu page.

- F03 Tagline
  - The "enjoy here or takeaway - the choice is yours" tagline appears on the main page and again on the menu page to let the user know that seating is available and to help re-inforce consistency in branding.

- F04 Company Ideals section
  - The company ideals are presented using 4 flip cards below the tagline on the main page. These cards give the user more information about the organisation in relation to its strategy in the areas of quality, sourcing of ingredients, recycling and philanthropy.

LO4.3 **Attribute all code from external sources to its original source via comments above the code and (for larger dependencies) in the README.**

Tip: Be transparent about any resources used in your project. Ensure to include it in comments, if a particular code was used, or in README if you’ve used something for research purposes

LO4.4 **Clearly separate and identify code written for the website and code from external sources (e.g. libraries or tutorials):**

- Be transparent about each piece of code that was borrowed from an external source, utilise comments and include the source in Reference section on README

LO4.5 **Organise HTML and CSS code into well-defined and commented sections:**

- Utilise comments to describe features or anything else you would like to communicate
- Organise your code

LO4.6 **Place CSS code in external files linked to the HTML page in the HEA element.**

LO4.7 **Place JavaScript code in external files linked to the HTML page just above the closing body tag.**

LO4.8

## Write code that meets at least minimum standards for readability (consistent indentation, blank lines only appear individually or, at most, in pairs):

- Utilise Beautifier to format your code as necessary
- Ensure consistency

LO4.9 **Name files consistently and descriptively, without spaces or capitalisation to allow for cross-platform compatibility.**

LO4.10 **Group files in directories by file type (e.g. an *assets* directory will contain all static files and may be organised into sub-directories such as *CSS, images,* etc.)**

- Ensure to sort the files by the file type and into relevant folders
- Keep the directory structure simple and easy to navigate

# LO5

**Demonstrate and document the development process through a version control system such as GitHub**

LO5.1 **Use consistent and effective markdown formatting to produce a README file that is well-structured, easy to follow, and has few grammatical errors**

Useful resources:

- Markdown-cheatsheet

# LO6

**Implement Front-End interactivity, using core JavaScript, JavaScript libraries or frameworks**

LO6.1

# Write JavaScript functions that correctly implement compound statements, such as if conditions and loops.

Revisit JavaScript Flow Control  If/Else Statements challenges and others relevant to your project topics

LO6.2 **Write code that intelligently handles empty or invalid input data.**

- Make use of the required keyword where appropriate
- The user is unable to submit invalid data, such as empty input or flipping more than three cards in the memory game

LO6.3 **Implement appropriate working functionality for all project requirements.**

- Manually test your application and ensure that all functionalities work as expected

LO6.4 **Write code that does not generate internal errors on the page or in the console due to user actions.**

- Ensure to check the console throughout the development process and if any errors are introduced, eliminate them
- Final version of the code should not generate any internal errors on the page or in the console

LO6.5

# Organise code and assets files in directories by file type.

# LO7

**Use an Agile methodology to plan and design a Front End Web application**

LO7.1

## Use an Agile tool to manage the planning and implementation of all significant functionality

LO7.2

## Document and implement all User Stories and map them to the project within an Agile tool