Started making logo on <https://www.brandcrowd.com/>

Edited logo to be circular using online tool - <https://crop-circle.imageonline.co/> after first extending canvas with https://www12.lunapic.com/editor/

Added content for home page and sourced images from <https://pixabay.com/> (copyright free)

Decided to use clickable images for both recipe of the week and the quiz.

Started styling homepage. Had difficulty getting things to sit and look right. Ended up not using the slogan as it didn’t look good. The logo was too big for the icon on the left of the header, so made a cut down version to use and used border-radius to round corners. Made website name central on header bar, looks better. Used flexbox to keep header formatted as window is scaled.

Removed body margin to take colours right to the edge of the screen.

Due to not using full version of logo as icon, wanted to add larger one top of homepage which would also add to the user flow. Image gets stretched (at least on my screen) if I try to adjust its max-height. Used with flex and then buffered out the background colour by filling out the section.

Originally had header as fixed, but the logo banner was getting a little lost behind it, so made it sticky instead which solved the problem. Could have made the banner relative, but then that affects everything else in the page.

Issues using @media, discovered was missing meta for viewport (not mentioned in lecture/lab from what I can see). Had to adjust some of my widths to vw as once viewport started working, my scaling was no longer responsive. A little confused as to if I should be using a fixed unit such as px or responsive such as vw. Research suggests vw, but requires further research to gain more understanding.

Used flexbox for nav bar. Allowing hyperlinks to be side by side on desktop, then on top of each other for mobile.

Added hover and active colours to the anchors in the nav bar.

Noticed text was fine on laptop and responsive view, but very small on mobile. So attempted to utilise further for mobile.

Made nav bar sticky, so it gets moved up until just under the header bar and becomes fixed.

Used flexbox for positioning images and elements around the page.

Sourced and cropped all images (except logo) to 1920x1280 (seems a common format to find copyright free online). This will allow me to swap out images with javascript later on without any positioning or formatting issues

Used hover for changing style for all anchors, probably should change to javascript if time.

Added logo to head for tab icon.

Researched how to read in a json file. Found following to import json file as json object

<https://www.youtube.com/watch?v=Z92PqSyUBSI&ab_channel=AllThingsJavaScript%2CLLC>

Started writing json file to contain recipe details to obtain all current recipes available or for certain cuisine/courses/diet ect. Allowing the website to load up what is available. If a recipe is added, only that particular recipes html file need be created and its details added into the json file. The html pages which link to it will not need adjusting.

Jquery -----------------------------------------------------------------------------------------------------------------------

Could be be parsed with jquery or fetch, but trying to avoid using external APIS.

Researched how to add jquery via a javascript file. This will allow updating the version if required only in one place rather than all files which use it.

Found function to add on stackoverflow:

<https://stackoverflow.com/questions/1140402/how-to-add-jquery-in-js-file>

Had a lot of difficulty getting json to work inside javascript. Had to be type module to use import, didn’t want to use jquery or ajax.

Couldn’t remove child node by class. So commented out for now.

Started getting ‘cannot use import outside of module’ when linking page which called the javascript module errors. That page worked fine on its own, only linking caused issues.

Also found could not call functions inside a module from the html page.

Thought of using local storage to store the json object after reading on home page within a module. Then this allows access from other javascript files and then other html pages can use the object and pass parameters to javascriipt functions.

Populated list pages using json object stored in local storage. Thought about doing one template recipe page and populating from json, but thought it might affect the history. Decided to create a separate html page for each recipe, this will stop the history being affected if the user was to click through multiple recipes and then hit the back button.

Now when a new recipe is added, its details should be added to the directory.json file under “recipes”. Its cuisine, course, and diet should be checked that they are present in the relevant lists. If not, a new item should be added for that cuisine/course/diet along with a relevant image and html page for that cuisine, course or diet. Then the recipes html page should be created. Nothing else should require changing and the list pages should automatically pick up the new recipe from the json file (ensure to have visited the home page where the json file gets loaded to local storage). Downside, feel like theres a lot of repeat code.

Could add recipe of the week to json file if time.

List pages created and images/links working correctly.

Started creating recipe page. Started by making recipe image and description a flexbox. Added dropdown select box for portion sizes. To be used by javascript later to change quantities.

Added javascript functionality for populating course/portion/times/ingredients from data in json obj.

Some of my functions are quite long, but no time to go back and split out to cleaner code.

Discovered that my site would work fine through VS Code, but due to using root paths, images and links were not being loaded correctly when I ran the site by opening up the html file or via GitHub pages. Had to modify paths and pass modifier to js file to correct for nested files calling same script.

Discovered browser was blocking the js file of type=”module” (also blocks import statements). This worked fine via VS code, but found error when pushed to github pages. Ended up having to load json object to local storage by creating json object in js file instead of importing from json file. (no time to look into further).

Set recipe of the week image, link and header now gets set from JSON object information.

Lamb tikka masala recipe adapted from: <https://www.greedygourmet.com/recipes-for-diets/omnivore/lamb-tikka-masala/>

Cooking unit conversions: <https://www.cookingnook.com/cooking-measurements-conversions/>

Added “recipe-required” to json. Now if a recipe requires another recipe as an ingredient, the link and quantity will be provided in the recipe and styled differently. See lamb tikka masala as example.

Radio buttons put in for units (no functionality as yet). Default is checked by what is in json obj in local storage.

Realised had started part making a landing page (storing all information in json obj and populating a single recipes page), but had also started creating a page for every recipe. Realised that to protect history and correct page being loaded when user navigates to multiple recipes via other and then hits the back button, to be able to reload the correct page, would need to store variables. So binned that idea and went back to one recipe html page for each recipe. But reduced the amount of info stored in json file (no more ingredients). Links to images and other required recipe pages would now be stored in json. The links (href and src) are now set through javascript getting information from json obj.

Wrote function to convert ingredient quantities when portion selector was adjusted. Found when I stored the elements after obtaining via getElementsByClassName, each time the elements were changed, it also affected my variable. So had to extract innerHTML from elements and store in a list variable. This way all values could be based on original portions and original quantities (avoiding any loss). However found after making a change and going back to the original, whole numbers were getting set to 2 decimal places (as I was outputting toFixed to stop long values being displayed). Found had to first cast to an integer so zeros get truncated before toFixed. Function does not currently check what the unit is, so things like garlic cloves can have decimals, when in practice they should only ever be whole units. Something to fix if time.

Wrote function to convert units. Store original quantities and units in global variables, which can be used to reset to avoid any losses when going back and forth between units/portions. All units and portions are reset to default each time the page is loaded.

Ran all html files through w3c validator. Input some of the image source paths and anchor paths to stop warnings. These will be replaced when the javascript runs on page load.

Started creating lots of recipes. Sources:

Curry base sauce - <https://www.greedygourmet.com/recipes-by-national-cuisine/english-recipes/curry-base-sauce/>

Img for curry base sauce - <https://unsplash.com/>

Vegetarian Margherita - <https://www.bbcgoodfood.com/recipes/pizza-margherita-4-easy-steps>

Pizza Dough - <https://www.jamieoliver.com/recipes/bread-recipes/pizza-dough/>

Images for pizza and dough from <https://pixabay.com/>

Image for Chinese chilli chicken from <https://stock.adobe.com/uk/>

Image for Chinese cuisine from <https://pixabay.com/>

Added function to check and modify the “units” to/from singular/plural when units or portions are changed, realised later that item names (e.g. onion(s), egg(s)) could also be modified when time. Some units are being converted to volume, when probably should be weight. Would require new category probably.

Added function to change temperatures and units when going from/to imperial/metric units

Page flashes sometimes, and can be slow on first load – I believe this is due to loading images via javascript. Have done this so as all file and image paths can be modified in one place (read-json.js – json\_info obj). However, it is fine the majority of the time.