Assignment 3

User Interface Development

INFO30009

Shiquan Zhang – Tutorial Tuesday 4pm

Rachel Chen

Racchen1@student.unimelb.edu.au

1464253

https://github.com/rachel-c000/UID_Assignment3_RachelChen.git

https://rachel-c000.github.io/UID_Assignment3_RachelChen

Project Overview

This project involved developing a working implementation of a responsive e-commerce website, this was based on the prototype created in Assignment 2 of this course. Assignment 2 included the process of re-designing the website (desktop and mobile applications) of an e-commerce spice company, Teeny Tiny Spice Company, for improved usability, user experience and user interface. This concluded in high-fidelity prototypes of the re-designed webpages that a normal user will use to purchase a product including the homepage, search page, product page and shopping cart page.

To build upon this, Assignment 3 focused on building a fully functional and published front-end for the online spice store based on the prototype design. This was done using front-end coding languages including HTML, CSS, and JavaScript in the program Visual Studio Code. Alongside VS Code, Git and GitHub were used for tracking and management of the commits and project versions as well as hosting the final website.

The pages coded included the ones developed in Assignment 2 such as the home page, browse/search products page, product page and shopping cart page plus additional task flow pages and minor changes. These adjustments are further improvements that were made apparent through the coding process of this Assignment. This includes modular implementations re-structuring of the design which would in result improve usability and user experience, this will be discussed later in the report. In conclusion, the aim of this project was to publish a completed and accessible online implementation of the re-design of Teeny Tiny Spice Company.

Changes from Assignment 2

Through the developmental process from Assignment 2 to Assignment 3, there were certain interactive elements that were too difficult to implement into code such as the spice carousel bar. This was changed to a much simpler design in the coded implementation where the carousel was not structured with three spice products in view and the centre one enlarged, instead each spice is displayed the same size.

Another change was re-structuring the photos gallery content to be more modular and efficient for users to view. The previous prototype design had the media spread irregularly throughout its section effecting accessibility and user flow *(see figure 1)*. Whereas the working implementation structures the gallery in set columns and rows which gives users the visual clarity required to have a good user experience *(see figure 2)*.

Figure.1.
Assignment 2
photo gallery.

Figure.2.
Assignment 3
photo gallery.

In addition to this, within the prototype of the product page, it did not list the price of the selected product, which is a negative pattern in the task flow for users (see figure 3). This is because they cannot see directly the price of the product and would have to click back to the product listing page to discover this. In the working implementation, the price of the product is displayed on the individual product page (see figure 4).

Figure.3.
Assignment 2.
Spice profile.

Figure.5.
Assignment 2.
Cart icon.

On top of this, where originally when users click the 'add to cart button' the quantity was added in a small red icon on the shopping cart icon in the navigation bar. However, this is ineffective for users as the navigation bar is not on the screen for them to see because of the restrictions to the screen size and the page length. Instead, a pop-up message indicating the item and the quantity that was added to the cart is displayed. With this, users can close the pop-up and continue shopping or directly access the shopping cart to finalise their purchase.

List of Code Files

1. **Index.html**
   The structure of the homepage of the website which includes the navigation bar, animated spice banner, introduction text, spice carousel, word card sections, and recipe carousels.
2. **Search-Result.html**
   The structure of the search result webpage, this includes carded product displays showing the product, its name, stars and price as well as suggested similar and other products related to a user's search input.
3. **Product-Page.html**
   The structure of the individual product page displaying media, content and text related to the product.
4. **Shopping-Cart.html**
   Most input containers, buttons and references to JavaScript for interactivity and calculation of products and prices.
5. **Payment-delivery.html**
   Mostly input containers taking user information for payment details and delivery details.
6. **Checked-out.html**
   Feedback confirmation of the product purchase and a hyperlink back to the homepage.
7. **Style-Sheet.css**
   Referenced in all the html files, styling for all the webpages.
8. **JavaScript.js**
   Referenced in various html pages that need interactive elements, supports the interactive and calculations of webpages.
9. **Spice-carousel.html**
   Specific file with all html, css and javascript for the spice carousel, is put into the index.html with iframing.

Critical Reflection

The project was successful in implementing the basic structure and user interface of the previous prototype developed in Assignment 2. It was successful in following the theme including the curated the typography, colour palette, icon usage and components specific to the redesign of the Teeny Tiny Spice Company. The implementation cognitively communicates to users the theme of the company, the products they sell and their intentions for giving spice variety and ingredient transparency. This report used tools such as ChatGPT for prompt and code generation.

However, there could be many more improvements to add in future changes to the website. In the assignment most difficulties included coding the interactive elements including the carousels, buttons, and cart calculations which were needed to be coded in JavaScript. In future, this would increase the user experience of the website making their task flow more customised and efficient.

Another improvement for future cases is the organisation of the code and files within VS Code. While working with the code when there was no clear path of organisation was a difficult task as classes would overlap each other, and CSS would clash. If code and files are organised specifically then coding would be more manageable. An example of this is having a single main CSS file where it imports all the core and shared styles (layout, typography, components) and then page-specific styles in subfolders.