University for the Common Good

School of Computing, Engineering and Built Environment

Web Application Development 1
Coursework Specification

2021/22 Trimester B

Module Code:
M2I325626-21-B
M2I326554-21-B

This module is assessed 100% by coursework.

The coursework is due by 25.04.2022 at 16:00

In this coursework, you will implement a web application using the technologies and concepts you have learned in this module. The goal is to develop a web application aiding a restaurant in its day-to-day business.

Web Application

The web application should follow a well thought out style and consistent design. It should consist of **FOUR** web pages whose content and filenames are specified as follows:

1 - Homepage (*index.html*)

The main page should serve as a landing page for users. You should include an image of a food item and general information about the application and its functions.

Additionally, the page should contain information about the developer/development company of this page. This should include a few lines of marked-up text about their background and experience and a picture. This information can depict a fictional person or company using a stock image or similar.

2 - Bill Calculator (*bill.html)*

This page provides the user with an order calculator based on items ordered within the restaurant.
There are a total of 5 menu items a user can choose from including three food items and two drink options. The user should be able to provide a text comment. They should also be able to specify if a tip is added to the order. They should be able to enter and select whether the tip is a percentage of the order value or a set amount.

The options for the order form are as follows:

Food:

- Garlic Bread £3
- Pasta £8
- Pizza £9

Drinks:

- Lemonade £2
- Juice £2.5

Comment

- User can provide a text comment

Tip

- Input field for a value with a selection for the percentage of combined food and drink order or fixed amount to be tipped

Upon submission, this form should ***dynamically*** update a separate section on this site showing the following in a well-formatted and styled output:

- The total amount of £ for Food
- The total amount of £ for Drinks
- User-Provided Comment
- Tip in % of the food and drink order value **and** in £

- Total in £ (including the tip).

Fewer marks will be awarded if the output is only shown in a JavaScript *alert*.

Example: User enters a food order of 3 Garlic Bread, 2 Pasta, 1 Pizza, 2 Lemonade and 1 Juice in addition to a tip of 20%.
The results section should show the total amount for food (£34.00), the total amount for drinks (£6.50) the tip (£8.10 /20%) and the total price: £48.60 as well as the comment the user provided.
Note: All values should be rounded to two digits after the decimal dot.

3- Food Alert (*alert.html)*

This page will provide users with food alerts from the Food Standards Agency.
As the *minimum*, the page should display the top 10 most recent food alerts including the title, date, business affected and a short risk-statement for each food alert. Additionally, it should also include a description of the content that is displayed on this page. This will aid the restaurant in deciding which items need to be discarded or replaced.

For this, you are required to use the Food Alert API from the Food Standards Agency:

- https://data.food.gov.uk/food-alerts/id : API Link
- https://data.food.gov.uk/food-alerts/ui/reference : API Reference

In detail, the user will open the page and see a well-formatted, dynamic overview of the 10 most recent alerts. Those alerts will have been retrieved, following a call to the endpoint with the URL: https://data.food.gov.uk/food-alerts/id?_sort=-created (This includes the sorting-by-date query). It should then display the following properties:

- A shortened version of the title of the alert.
- The date the alert was created.
- The name by which the business affected is commonly known.
- Text describing the problem in terms of the risk to consumers for the *first* problem (if multiple exist)

The exact JSON object properties for the required data can be determined from the API documentation: https://data.food.gov.uk/food-alerts/ui/reference

More marks can be gained by including additional information and adding more functionality to this page. The requirements for this are:

- Adding interactivity to each search result. All items should be collapsed by default and only show the alert title. Upon clicking, users should be able to see all the other details for each food alert.
- Providing information about the 10 most recent items but also providing the user with a search option above the list of recent items. This option queries the API based on keywords and works in *addition* to the default list of 10 items. Upon using this search function, the recent items should be replaced with the top 10 results from the search, with data being received from the following API endpoint:

https://data.food.gov.uk/food-alerts/id?_sort=-created&search={...} (with {…} being replaced by the search query)

- Including more information from a *second* API endpoint to each search result. This is implemented by building an URL query to the following URL (http://data.food.gov.uk/food-alerts/id/{...}) and using the "notation" property of the alerts as the id instead of the "{…}". From the result, users should be able to see the "actionTaken" and "consumerAdvice" property values.

> ***Note*:** The focus for this part is on integrating a second API call. If the information is only received using one API call through the "_view=full" URL query, **no marks will be awarded for this**

4 – Order & Info (order.html)

The Order & Info page should contain an order form to order more supplies for the restaurant from their suppliers.
Fields for this order form should include the following fields:

- Person Ordering (Name) Field
- Contact Address (Email) Field
- Order Details (This will be a multi-line message) Field
- Supplier Choice - Users should choose *one* out of the four possible suppliers:
  - Argyll Butchers Ltd
  - Benson Bakeries
  - Caledonian Fruit and Veg Supplies
  - Daisy Fresh Dairies
- Shipping Option ("Express-Next Day" or "Standard – 2-3 Days").

This form does not need to be connected to any kind of database. As this module did not cover connecting this, it is sufficient to display the submitted data in a JavaScript Browser *alert* once the user clicks on the Submit button (replace the content in square brackets with the user-submitted values):

```
Person Ordering: [submitted Name]
Contact Address: [submitted Email]
Order Details: [submitted Order details]
Supplier Choice: [submitted Supplier Choice]
Shipping Option: [submitted Shipping Option]
```

This page should also contain a description of the form and what it is used for in the restaurant.

Other Web Application Criteria

In addition to the aforementioned functions of the web application other criteria have to be met as well:

- The web application should be responsive and should be designed to display across a range of different platforms (at least mobile and desktop)
- Sufficient navigation options should be provided such as links between the different pages and internal links (if used).
- For all input fields, sufficient warnings should be added if the user does not enter correct data. This means that when the user enters text into a field that is designated for numeric values or the user enters something which is not an email into an email field, an appropriately styled warning should appear
- You should use a user-friendly colour scheme. Some inspirations for this can be drawn from pages like: https://coolors.co/, http://colormind.io/, https://designs.ai/colors.
- You should **NOT** use sample text; your web application should serve as a finished product.
- The use of **Content Management Systems** such as WordPress, Drupal, Joomla or prewritten frameworks is **NOT allowed.**
- Code libraries such as jQuery and Bootstrap are allowed.
- You should use semantic-markup as appropriate (<main>, <header>, <footer>, <section>, <figure>, <article>…..)
- External Stylesheets and JavaScript files *should* be used
- Styles should use different kinds of selectors, in particular
    - Element, Class, ID and Pseudo-Selectors
- All code files **must** include sufficient comments. It is not necessary to comment on every single line of code but larger- or more complicated code should include them (This also applies to CSS)
- The web application must work correctly in *Chrome* or *Firefox*.
- All HTML and CSS files must be validated. The results of the validation for **each** of the four pages as well as the validation report for CSS used need to be included in the submission. Use the validators at:
    - https://validator.w3.org/
    - https://jigsaw.w3.org/css-validator/

  Take a screenshot of the validation reports, label it accordingly such as "*validation index.jpg"* and save it within your submission.

Plagiarism

You should pay attention to the university's codes and practices[1] as well as their plagiarism regulations[2].

[1] https://www.gcu.ac.uk/academicquality/regulationsandpolicies/universityassessmentregulationsandpolicies/

[2] https://www.gcu.ac.uk/library/smile/plagiarismandreferencing/

Any kind of content (images, text, ***code***) that was copied from any source and used in your coursework **without** acknowledging the source is bad academic practice and could fall under plagiarism.
The discussion of coursework between students is encouraged but the work has to be undertaken individually. ***Collusion*** (copying work between students) may result in a zero mark being recorded for everyone involved and further action being taken.

Sources

To avoid plagiarism or poor academic practice you need to ensure that you specify where you obtained any material you use and how you have modified it.
This **MUST** include specific web addresses (not just google.co.uk).
A template for this is included on GCU Learn and an example is shown in Table 1.

Table 1 Sources Template

Type	Sources
CSS	Table Styling https://support.awesome-table.com/hc/en-us/articles/115001399529-Use-CSS-to-change-the-style-of-each-row-depending-on-the-content
JavaScript (modified)	Splitting String in split_string JavaScript function: https://stackoverflow.com/questions/96428/how-do-i-split-a-string-breaking-at-a-particular-character
Image	GCU Logo https://www.gcu.ac.uk/

Coursework Submission

All files for your web application should be submitted in a compressed *.zip* file. The file should follow the following naming scheme: LastName_FirstName_StudentID.zip.
If your name is *Nicola Sturgeon* with the StudentID *S12345*, name the file:

sturgeon_nicola_S12345.zip

Coursework files are submitted using GCU Learn.

You should double-check your files before submission to ensure that you did not miss any files.

Marking Scheme

The marking scheme which will be used to assess the coursework is appended below.

Practical Based Web Programming Assignment Total: 100 Marks	100 % of overall coursework mark				
Topic	Fail <40%	3rd 40%-49%	2.2 50%-59%	2.1 60%-69%	1st >70%
Visual Design 10 Marks	No attention to the visual design of the pages of this application and the placement of information. No colours, or inadequate colour scheme used.	Little attempt at providing a coherent and consistent interface style. There is little evidence of attention to layout and consistency of the visual design.	Appropriate layout and design of information within the pages of this application.	Good layout and design of the information within the pages. The visual design appears consistent and there is a clear differentiation between different kinds of screen sizes.	Excellent visual design of the web application. The design appears appropriate, clear and consistent and fits to the specification. There is a clear differentiation between different kinds of screen sizes.
Site Structure and content 10 Marks	Inadequate navigation or structure of the web application. Major omissions or placeholder content included.	Poor structure of the application and its content. The navigation does not appear to be thought out and may include occasional omissions and placeholder text as content.	Reasonably structured web application. Necessary navigational tools provided. All necessary areas included and no placeholder content.	Good, well thought out structure. Clear and consistent navigation between the different web pages of this application.	Excellent, clear and concise navigation and structure. The pages are very user friendly and adhere to modern conventions allowing users to find content and use them with ease.
HTML Markup 20 Marks	No real attempt at using HTML markup. Errors are present in the markup and deprecated features have been used.	Some attempt at using the appropriate markup for the web application.	Structured markup used to create an adequate site. Shows use of relevant semantic elements.	Good attempt at integrating the correct markup. Semantic elements have been used throughout the application and it is built with regards to current standards.	The markup of the pages of this application is excellent. Complies with current standards and is built with regard to future development. Consistent use of the relevant HTML elements using well-structured code.
CSS 15 Marks	Inadequate attempt. Content and layout are not properly separated, very little styling has been applied	Very simple rules, some selectors and common properties used to create a very basic layout	External stylesheets used. Different rules for different elements have been applied to create a straightforward layout. Multiple types of selectors have correctly been used to apply a straightforward layout	External stylesheets have correctly been applied. A variety of selectors, properties and values such as relative units have been used to create an appropriate layout. Application shows use of media queries to create a responsive design.	Well-structured, external stylesheets have correctly been applied to create a responsive, original and well-designed web application. Styles have been applied to work well on different device-sizes, e.g. are responsive.

				Evidence of some originality and creativity.	
Client-Side Functionality and Interactivity 15 Marks	No real attempt at providing interactivity and the required functionality using JavaScript	Occasional Flaws in a straightforward attempt to provide some level of required functionality	The client-side functionality demonstrates reasonable understanding and competence in the technologies taught. Minor flaws or omissions may be included.	Good attempt at providing the required functionality	Excellent client-side functionality covering all the required specifications with well-structured and well-designed code. Student demonstrates a very good understanding of the technologies chosen.
API-Access 10 Marks	No real attempt at accessing external data through APIs.	Little attempt at using JavaScript or HTML5 to receive data from external resources. Or Data is fetched from the API but not integrated into the webpage	Appropriate attempt at using JavaScript or HTML5 to fetch data from remote APIs and integrate it into the webpage	JavaScript or HTML5 has successfully been used to fetch data from remote APIs and display them appropriately	All relevant data from the remote APIs has been fetched and is displayed in a structured, well designed, engaging interface
Project Scope 15 Marks	Inadequate attempt at matching the coursework specification.	Little attempt to provide the appropriate levels of interactivity and functionality of this application.	Reasonable attempt at providing the appropriate levels of interactivity and functionality with some features missing or erroneous.	Good attempt and fulfilling the project specification with little flaws.	Excellent well-developed application which clearly fulfils the project specification
Validation & Distribution 5 Marks	No Real Attempt. Validation errors present, files missing, incorrect URLs used.	Occasional errors in the distribution which is generally adequate. Some validation errors present.	Adequate attempt at file organisation and little to no validation errors	Good file organisation, suitable file formats used and mostly no validation errors.	Well organised site, showing high level of file organisation and no validation errors of HTML and CSS files.