cct | College Dublin
Computing • IT • Business

Programme Title/Year:	**BSc Year 2**
Module Title:	**Client-Side Programming / Programming: OO Approach**
Lecturer Name:	John Snel / Ken Healy
Assignment Title:	Interactive website
Assignment Type:	Continuous Assessment 3 (CA3)
Assessment type:	Individual
Weighting:	**Client-Side Programming -** 40% **Programming: Object-Oriented Approach - 10%**
Date issued:	
Due Date:	**15/05/2021, 23:55**
Method of Submission:	Submission **must be** uploaded to **Moodle**. No submission will be accepted via email.
Late submission:	A penalty of 10% of the marks awarded is deducted for any work that is submitted after the deadline and up to 5 working days late, after which a mark of 0 is awarded.

Module Learning Outcomes

For this assignment, the learner will be assessed on the following module learning outcomes:

- **MLO 2 -** Write well-structured client-side code for interactive web applications
- **MLO 4** - Integrate asynchronous data processes to develop and/or maintain an application
- **MLO 5 –** Evaluate and choose between third party libraries and frameworks for client-side development with regard to specified requirements

Instructions

You are required to create an interactive website with two pages using a combination of HTML, CSS and JavaScript.

Page 1

For this page, you are going to create a web application for a restaurant to calculate the cost of at least 5 customer orders.

1. **Staff password validator:** create a password validator, which tests whether or not the staff's desired password is strong enough. The criteria which must be met in order for a password to be strong enough are as follows:

a. The password must have at least one lowercase letter.
b. The password must have at least one uppercase letter.
c. It must contain (at least) one digit and (at least) one special character.
d. And it must be at least eight characters long.

2. **User[1] (customer) generation:** you are required to create at least 5 randomly generated user (customer) profiles for the customers using JSON data returned from the random user API. You can use the random user API, https://randomuser.me/ to get data for the customers. Your solution should dynamically generate image thumbnails for the customers.

3. **Food cost:** the application should include a form that will allow the catering staff member to choose from a range of available menu items. The menu should include starters, main, dessert and drinks. There should be a minimum of 3 items in each category (i.e. 3 starters, 3 mains, etc.). Some of the guests may be vegetarian, so the starters and mains should contain at least 1 vegetarian menu item. Each menu item is an object containing: *the dish name*, a short (1 line) description, *a suitable variable* that indicates whether the dish is vegetarian, and *the cost of the dish*. A breakdown of the bill will be returned to the catering staff member showing:
    a. the overall bill
    b. a summary of each of the costs subdivided by starters, main, dessert and drinks.
    c. a comparison of the cost of the vegetarian meals ordered vs the cost of the non-vegetarian meals–this only needs to be calculated on the starters and mains.

Page 2

For this page, you are required to research and develop interactive components on this page to showcase your knowledge of JavaScript and demonstrate your ability to integrate **any** third party library and/or frameworks. There is no restriction as to how many Javascripts features you can use here, except that this page should include something that is related to page 1, for example, related to food, restaurants, or the country from which you have chosen the cuisine. The purpose of this page is for you to showcase your ability with Javascript and a framework.

Submission

Your submission should be uploaded to moodle and you are also required to put your project on your personal **Github repository.** This repository should show **an adequate number of commits** that go back to the start of your project.

[1] Note we have left it as user because it is called a random 'user' api but this is referring to the customer profiles.

cct | College Dublin
Computing • IT • Business

Marking Scheme - Client-Side Programming

(Weighting - 40%)

Description	Weighting
Page 1: Staff password validator. [Validation works correctly and checks all the rules set out in the requirements; An appropriate error message is displayed for invalid data]	10 Marks
Page 1: User generation. There are at least 5 random users (customers) generated and thumbnail images for each have been generated	15 Marks
Page 1: Food cost - The catering staff member can choose from a range of options for starters, mains, desserts and drinks; there are at least 3 options for each; at least 1 vegetarian option exists for starters and mains; The catering staff member is presented with a total cost/total bill when finished selection as well as a breakdown of the cost for each category	25 Marks
Page 2: Demonstrable use of JavaScript	15 Marks
Page 2: Demonstrable use of a Library/Framework	25 Marks
Use of Github: There are a series of logical and coherent commits over time showing the development of the project **[NB: If you make just 1 commit, or a number of commits in the last few days before the deadline, you may score zero for this element!]**	10 Marks
TOTAL	**100**

cct | College Dublin
Computing • IT • Business

Marking Scheme - Programming: Object-Oriented Approach

(weighting - 10%)

Description	Weighting
Quality of Programming Documentation [Javascript has been properly commented and code has been explained clearly; variables have been named appropriately and in line with Javascript conventions]	25
Appropriate use of Objects/Classes [Classes/Objects have been named and constructed following good programming practice]	10
Neat Code [Javascript code has been properly indented and is easy to read]	10
Quality of Javascript code / Use of Functions [Javascript code (created by the student) demonstrates an understanding of good programming practices; functions have been used appropriately to modularise code]	30
Level of Programming Challenge [This will be marked based on the level of difficulty of any code created by the student]	25
TOTAL	**100**