INTRODUCTION

The interactive web (Web 2.0) has changed the way people use the internet. Creating powerful webpage functionality and interaction through JavaScript requires a deep understanding of the language's data structures, algorithms, and libraries.

This term will focus on these deeper concepts as well as tools and libraries that support this level of intermediate development.

PROJECT OUTLINE

Create an interactive website for a coffee shop

Background and Context:

The website you will be creating is aimed at allowing users to order coffee from the shop. A single order is made up of many coffees. The order must be processed to determine the name and price of each coffee and the total price of the order.

What you will do:

You are in charge of creating the website for the coffee shop. This website will be made up of multiple HTML pages, CSS stylesheets, and JavaScript files. You must create all these files yourself.

The website should contain the following pages, created manually by you:

- Home page
- Order page

The home page must contain:

- A navigation system
- The name of your coffee shop

- A short description of your coffee shop
- A carousel of images

The order page must contain:

- A form through which users can order coffee
- A summary of coffees ordered so far, with their names and prices
- The total price of the order so far

The business logic of the coffee shop:

- Each coffee is made up of up to 5 ingredients:
    - Espresso
    - Cream
    - Milk
    - Sugar
    - Syrups (caramel or hazelnut)
- There are 3 sizes of coffee: Grande, Tall, and Short
- There are 3 coffee styles: Americano, Cappuccino, and Latte
- Any coffee can have extra ingredients added
- The name of a coffee is determined by its size, style, and extra ingredients
- The price of a coffee is determined by its size and the price of each ingredient

You must support this website with dependencies managed through NPM. These dependencies must include at least:

- jQuery
- Jasmine

jQuery should be used to add functionality through user-triggered events.
Jasmine should be used to create a unit test suite which tests the following:

- New coffees can be added to the order
- Coffee names are correctly generated
- Coffee prices are correctly calculated

Do some research to find a dependency that will allow you to easily add a carousel to your home page. Read the documentation to ensure you understand how to use the dependancy.

In addition to the website, you will also produce a rationale document detailing your design decisions and reasoning. You will present the website and rationale deliverables to your lecturer at the end of the term.

What you will be given:

You will receive a supporting document detailing coffee and ingredients from your lecturer.

PROCESS

This section of the brief contains helpful guidance and information concerning your process. You will be technically guided through each phase of the project.

Phase 1: Planning

Website planning

Design and plan the website. Consider the layout and navigation system so that users can easily use the website and its functionality. Record all assumptions, decisions, and reasoning in your rationale.

Form planning

The main goal of the website is the order functionality. Design your form to aid the user adding to the current order, preventing invalid inputs when possible, providing validation and feedback about errors, and displaying processing output in a clear and meaningful way. Record all assumptions, decisions, and reasoning in your rationale.

Phase 2: Implementation Phase

Website implementation

Create the home page and order page based on the designs from your planning phase. Make sure that the navigation is functional. Record the changes you made and what you learned during this phase in your rationale.

Object implementation

Create the JavaScript models and algorithms necessary to capture the website. Create the constructors for Order, Coffee, and Ingredient objects. Employees get a 25% discount when ordering from the store. Create the constructor for an EmployeeOrder object which inherits from the Order object. Update the prototypes of these objects by adding methods to determine names and calculate costs. Record the changes you made and what you learned during this phase in your rationale.

Event Implementation

Add events to your site using jQuery. These user-triggered events should cause objects to be created and methods to be invoked to produce output to be displayed on the order page. Any input errors should be reported to the user. Make sure to test this functionality thoroughly. Record the changes you made and what you learned during this phase in your rationale.

Phase 3: Testing phase

Automated Unit Testing

Ensure that your calculations are working as expected by creating a Jasmine unit test suite and ensuring all tests are GREEN.

Manual Testing

Ensure that your events are triggering correctly and that your output is displaying correctly through manual testing.

Validation and Feedback

Make sure that the order page communicates information to the user effectively. Any input errors must be considered and communicated, and users should be guided through the order process.

Quality Control

Make sure the pages on your website are consistent, follow your designs, display all the information that they should, and support all the functionality that they should.

Phase 4: Final Phase

Documentation

Go back to your rationale document and polish it. This document should include all your assumptions, decisions, and reasoning.

Presentation

You must present your deliverables to your lecturer and the class.

DELIVERABLES

Website files

The project folder including all your HTML, CSS, and JS files as well as your `package.json` manifest file. Do not include your `npm_modules` folder.

Rationale Document

This document should be delivered as a PDF. It must include all your assumptions, design decisions, planning and reasoning.

In addition to the soft copy you submit to the Google Classroom, you must also submit a printed hard copy before presenting your project with an attached and signed plagiarism form.

Screenshots and Recordings

You must supply 4 large PNG screenshots of your project's output. Do not include screenshots of code. These files should be named `IDV202_xxxxxx_S1.png`, `IDV202_xxxxxx_S2.png`, `IDV202_xxxxxx_S3.png`, and `IDV202_xxxxxx_S4.png` where `xxxxxx` is your student number.

You must supply an AVI screen capture of you interacting with your application. The video should be no shorter than 60 seconds and no longer than 3 minutes. This file should be named `IDV202_xxxxxx.avi` where `xxxxxx` is your student number.

These files are going to be used for display and advertising purposes, so they should demonstrate the best elements of your project without being too technical. They should also include your name to help identify you while on display.

These files should not be included in your Google Classroom archive folder submission. Instead, you must add these files to the `IDV_SUB/202/` folder on the desktop of the lecturer's machine in A2 during week 9.

SUBMISSION

- NO LATE SUBMISSIONS WILL BE ACCEPTED
- Submission and presentation will take place in Week 9
- Projects should be digitally submitted on Google Classroom
- All deliverables must be placed in a single zipped archive
- The zipped folder must follow the naming convention **`xxxxxx_IDV202.zip`** where **`xxxxxx`** is your student number

OUTCOME

By completing this project, learners will have learned to:

- Create objects through constructors
- Manipulate object prototypes
- Add funtionality through events
- Automate unit testing
- Manage dependencies
- Document a website implementation
- Professionally present projects
- Manage their time

ASSESSMENT

Learners will be assessed on their:

- Data analysis abilities
- Planning
- Problem solving skills
- Decision making abilities
- Tool use
- Technical skill
- Technical understanding
- Correctness in implementation
- Usage considerations
- Device and hardware considerations
- Initiative and brief expansion
- Layout, communication and aesthetics
- Documentation
- Professionalism

**Please see the attached marksheet for a detailed breakdown of the assessment weightings.*

SCHEDULE

WEEK	DATE	CLASS WORK	MILESTONES
1	2017-04-17 to 2017-04-21	• Contact Week	• Contact Week
2	2017-04-24 to 2017-04-28	• Brief • Object Constructors • CLI: navigation	• Planning Phase
3	2017-05-01 to 2017-05-05	• Object Prototypes • CLI: files and folders	• Planning Phase
4	2017-05-08 to 2017-05-12	• Object Inheritance • CLI: editing files	• Implementation Phase
5	2017-05-15 to 2017-05-19	• Data Structures and Algorithms • CLI: searching	• Implementation Phase • **Homework assignment 1 due**
6	2017-05-22 to 2017-05-26	• Testing with Jasmine • CLI: package management	• Implementation Phase • **Progress milestone 1**
7	2017-05-29 to 2017-06-02	• Integrating with jQuery • Class Activity • CLI: dependency management	• Testing Phase • **Homework assignment 2 due**
8	2017-06-05 to 2017-06-09	• Constructing a rationale • CLI: networking	• Final Phase • **Progress milestone 2**
9	2017-06-12 to 2017-06-16	• Submission, demonstration, and assessment	