2 - Asynchronous Programming

1. Learning Outcomes

On completion of this lab you will have learned how to:

- Implement an interactive user interface in JS using event handling
- Process JS objects and arrays (from JSON) using the functional style
- Read and process remote JSON documents using promises
- Implement a simple networked-UI using DOM API first principles

2. Organisation

Please complete the exercises individually.

3. Grading

This worksheet is worth up to 10% of your overall module grade. You must attend and sign in at a minimum of 10 labs in the semester in order to obtain full CA credit.

You may work on this worksheet during lab 3 and lab 4 with instructor assistance. You may also be requested to demonstrate your submission to the lab instructor in order to receive credit.

4. Submission

The deadline for submission is Sunday Oct 21, 2018 @23:59 through Github.

The work and submission workflow is as follows:

- Before you start working on your worksheet you must **fork** a copy of the official class repo from here
    - https://github.com/bjg/2018-ditcs-cmpu4043.git
- Then **clone** the forked repo to your development machine
- The make a new branch in your cloned, local repo named as follows:
    - ***<student-id>***-wks-2

  where ***<student-id>*** is something like C12345678
- When you are finished developing your worksheet solution then you must push your local repo to the remote origin for that branch
- Finally, when you are submitting your solution for grading, you will generate a pull request (PR) requesting that your branch is merged with the remote origin master branch of the official class repo above

- If you are not sure about any of the described steps here, then take a look at this worked demonstration:
    - https://www.youtube.com/watch?v=FQsBmnZvBdc

- UPDATE: As some of you have pointed out, the instructions above only work for public Github repos. If you want to use a private repo for your working copy, then take a look at the method described here: https://help.github.com/articles/duplicating-a-repository

5. Requirements

For this lab and the remainder of the module, you will need to have created an account with Github (https://education.github.com/) which you can do so for as a student on their educate program

As subscriber on their education program, you will have free private repos. If you do not store you code in a private repo, you run the risk of someone else discovering your work and plagiarising it for themselves.

6. Resources

You are free to research whatever you need to solve the problems in this lab. Some recommended resources include:

- https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://www.codecademy.com/learn/javascript

7. Problem Sets

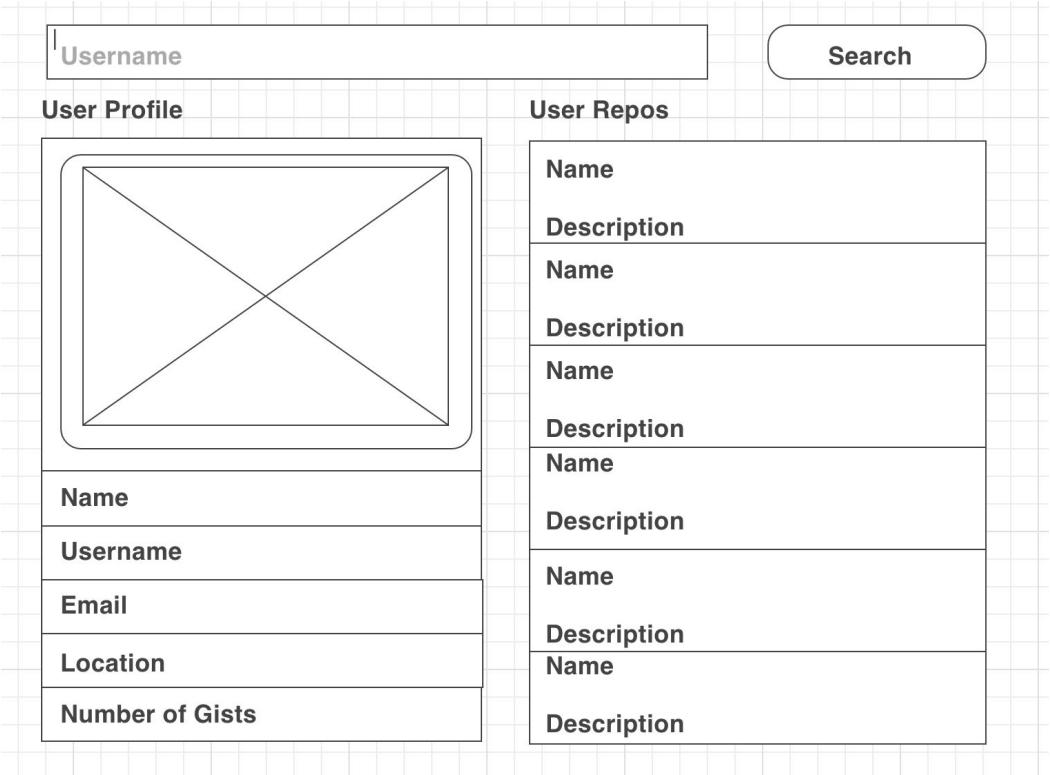
Provide Javascript ES6 code for the following problems using JS Bin or your own development environment as you prefer.

When submitting, clearly identify what code solutions are associated with which problems.

Note: Keep your JS Bin URLs private to you. Do not share these with class colleagues.

1	Add button and keyboard handling logic in Javascript for the calculator front-end you built and submitted from worksheet 1.	30 Marks

	Your solution should not use third-party libraries such as JQuery. Use only the native API - HTML5 or JS whichever you prefer. The final solution should implement a fully working calculator which will allow a user to use the mouse or keyboard to enter expressions and have them calculated and displayed in the front-end provided **Note**: You may edit HTML/CSS code that you have previously developed	
2	For this problem refer to the lecture on array operations and consult the documentation at: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API The API you will work with is here: http://jsonplaceholder.typicode.com Using this API and the functional programming style of array operations, provide code to solve for the problems below. You **MUST USE** this FP style with functions such as map, reduce, filter, flat and so on. If you find yourself using loops, you're doing it wrong. Carefully think through which data transformations you need to apply in which order in your pipeline. For each answer, use **`console.log()`** to display the data (i.e. you do not have to build DOM objects): 1. List of objects having the following **user** attributes: a. Username b. City c. Zipcode 2. Show the number of **users** having only zipcodes starting with the number 2 or the number 5 3. List all of the **post** titles having more than six words 4. Show a word frequency map for all of the body contents of the **posts**	20 Marks
3	You will build a web application for querying user information from the Github API that looks like the following:	50 marks



Notes:

- Use only native APIs, Javascript, HTML and CSS (no external libraries permitted)
- You may use whatever style of async programming primitives you like such as callbacks, promises or streams.
- The API endpoint for retrieving user information is https://api.github.com/users.
- The user of your application can input a username and search in Github for that user's information
- The information should be displayed as shown, including the avatar picture at the top left
- The repo information can be obtained by following the "repos_url". You should implement a scrollable list to display these if the number exceeds 5
- Marks will be awarded for code quality such as formatting and how DRY your code is