IMY 220
Assignment 5:
jQuery + AJAX

Due: Wednesday 23 September at 13:00.

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- Use jQuery and getJSON to add a simplified "infinite scroll" to a page of users.
- Download *index.html, users.json, bg.png, and loading.gif from ClickUP*. The index page contains information cards for three users and a loading bar at the bottom. The JSON file contains user details for three users.
- You must include a CDN version of jQuery and write all the code for this assignment inside a file called *script.js* which you must create and include inside *index.html*.
- You must use .on() for event handling and you must use the recommended syntax from the slides for adding new elements. Use the most appropriate keywords for declaring variables.
- In order to clarify the required functionality for this assignment, an example file called *A5_memo.html* has been included for you to download, which demonstrates what you are required to implement. (Don't worry about the code contained inside this file, just put it inside the same directories as the other A5 files and it use it to understand the required functionality. If your code looks anything like the code inside this file, it will not be marked.)

Section 1 – Scroll to bottom

Attach an event handler to the appropriate element that will fire when the user scrolls the page. Use this to check if a user has scrolled to the bottom of the page and log a message to the console stating that the bottom of the page has been reached. (If the content-load functionality for sections 2 and 3 don't work as required, you can still get marks for this section.)

Section 2 – Load User

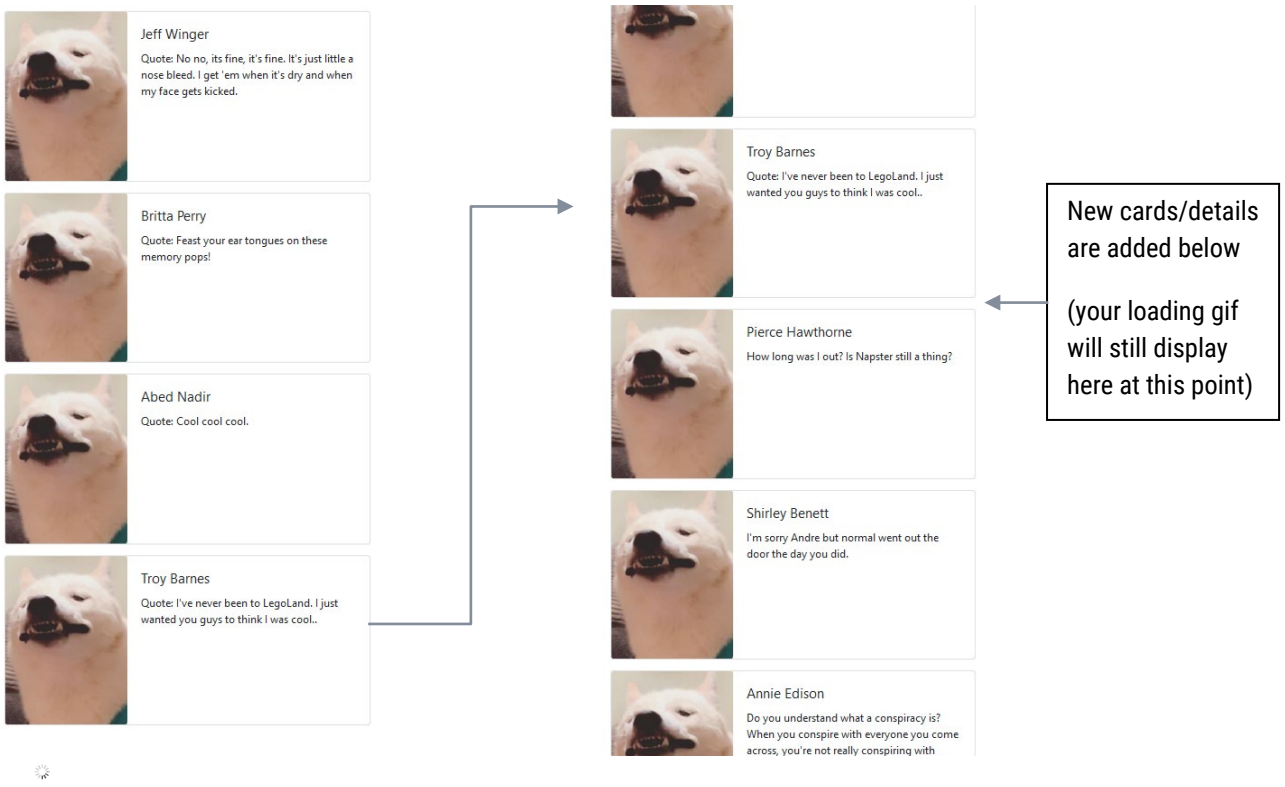
Define a function called getUsers which must take a URL as its only parameter and return a Promise. The Promise must use jQuery's getJSON function to load a JSON file which is specified by the URL-parameter. Also define a function called createUserCard which creates and returns a new set of elements for creating a Bootstrap card similar to those found inside *index.html*.

When the user scrolls to the bottom, use getUsers and createUserCard to create a new Bootstrap card for each user inside *users.json* and display them on the page. The result of this should be that when a user scrolls to the bottom of the page, new cards are dynamically added after the existing ones for the user details inside *users.json*, which is illustrated on the next page.

Note that you can ignore the loading gif for now, we will implement that in Section 3.

Before scrolling:

After scrolling



Section 3 – Loading indicator

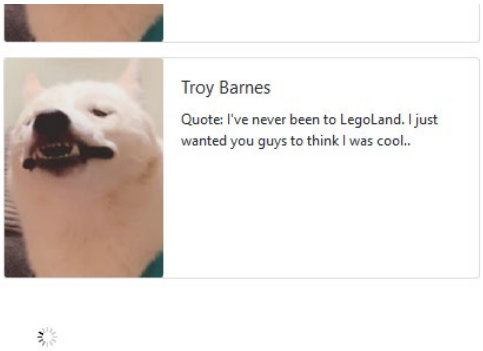
Create functionality which simulates a loading gif that displays while the details are being asynchronously loaded. (The reason why we are only simulating the loading of contents is because the script and resources are all found on the same server and are relatively small, which means that the contents will load too quickly for us to properly test the loading of external resources.)

To simulate the loading of resources, which on a real server could take a couple of seconds, we will make use of a timeout function wrapped in a Promise, the code for which is as follows:

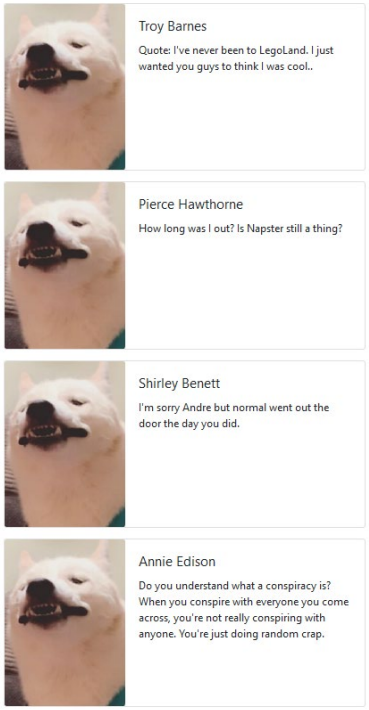
```
const sleep = duration => {
        return new Promise((res, rej) => {
                setTimeout(res, duration);
        });
}
```

You must use this function to pause the execution of the functionality for Section 2 for 1 second, in other words, when scrolling to the bottom of the page, 1 second must pass before the user cards are created and added for new user details. Note that you **must** use await for this purpose. You must also remove the loading gif from its position and attach it to the bottom of the newly added elements, for example:

Before scroll:



After scroll:



Additional Information

- Refer to the slides and online resources for help

Submit only the following file(s) according to the submission instructions.

- *index.html*
- *script.js*