Week 2 - Session 2

Exercises

1. Work through the following sections on https://javascript.info/
    - Part 1;
        - 9.2 - 9.4 (Further Classes)
        - 10.1 (Error Handling)
        - 11.1 - 11.5, 11.8 (Promises)
    - Part 2;
        - 2.1 - 2.2 (Events)
        - 3.1 - 3.2 (Mouse Events)
2. Create a basic page which displays a joke of the day from the Joke of the Day API. Instead of using `XMLHttpRequest` like their documentation, use Axios.
3. Go back through previous exercises and make them look nicer using Bootstrap - for example, the search form from the previous workbook.

Project

Scenario

Good news - Initech were impressed with the demo page! The only thing they commented on was that it was a bit "rough around the edges". They've requested that we remake the page, but this time using Bootstrap.

They've also given us access to an API they built which supplies mock check-ins. We need to demonstrate that we're able to integrate with their data directly, so you'll need to replace the hard-coded check-ins we're already showing.

You may include any external libraries via their CDNs - there's no need to download anything at this stage.

Useful Reading

- Introduction to Bootstrap (30 minutes) - Includes a very basic template to get you started
- Bootstrap's Carousel Documentation (30 minutes) - Demonstrates how to use a Bootstrap carousel
- Bootstrap's Grid Documentation (45 minutes) - Documentation and examples of column layouts for different screen sizes
- JSON Tutorial (30 minutes) - Explanation of data types in JSON
- Introduction to jQuery (1 hour) - Basic concepts of jQuery
- Getting Started with Axios (45 minutes) - Introduction to making AJAX requests with the Axios library
- Introduction to APIs (1 hour) - Summary of RESTful API resources

Tasks

1. Recreate the initial page (☕☕)

Create an HTML page which mirrors the design our designer has put together in `desktop.png`. This time, it should use Bootstrap classes. You may still use a static image for the star rating. This time, the page **must** be responsive.

Hint:

You shouldn't need to write any CSS here - Bootstrap classes will do everything that you need to do! The grid classes will allow you to alter the layout of columns for different screen sizes.

2. Replace the gallery with a Bootstrap carousel (☕)

The gallery thumbnails are no longer required - simply replace the gallery in the original layout with a Bootstrap carousel.

3. Implement the mock check-in API (☕☕☕)

Clone the mock check-in API with the following command:

```
git clone git@bitbucket.org:deeper-learning/mock-checkin-api.git
~/projects/mock-checkin-api
```

This will download the API into your projects folder. Within there, you'll find a readme file with instructions to get it running.

Update the page from the first task to implement the supplied API which provides mock check-in data. Instead of hard-coded check-ins, this time you'll need to display them from the data retrieved via the API. At this stage, you may display a number for the rating (ie without the stars).

Hint:

Create a div with an ID to contain all check-in information, and use jQuery to append elements to it dynamically.

What happens if there are no results returned?

4. Stretch goal: Display an aggregated rating at the top of the page (☕)

Replace the hard-coded star rating at the top of the page with an average rating based on the check-ins retrieved via the API.

Hint:

Calculate the mean of all retrieved ratings and convert it to a percentage, similar to exercise 4.

Further Reading & Useful Resources

- Bootstrap Documentation
- jQuery Documentation
- Axios Documentation
- JSON Schema