CS5002 Assignment 4: Data and the DOM

Due date: Thursday 26th November 2020, 9pm

40% of the overall module grade

You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.

Aim

The aim of this assignment is that you should be able to

- Load JSON data from a local file
- Use the data to build up a table on a webpage
- Use the select element to filter the contents of the table
- Add some interesting extensions of your own invention

Introduction

This is the fourth piece of coursework for CS5002. It involves loading and working with external data, displaying the data, and allowing the user to filter that data. It is divided into a number of requirements, that should be attempted in order, plus a set of suggested optional extensions.

Note I will be testing your submissions in Firefox version 80.0.1. So, while you may use any web browser, I recommend using Firefox to avoid any confusion with different versions of JavaScript.

Requirements

In this assignment, you will develop a JavaScript program that allows the user to analyse data about American movies released since the year 1900. The data was initially scraped from Wikipedia, and is available in the file `movies.json`. The data was downloaded via https://github.com/jdorfman/awesome-json-datasets, which, incidentally, is a collection of several interesting JSON data sets.

You are provided with `movies.json`, and you can also find a more nicely formatted file, `movies-formatted.json` and a cut down data set for testing purposes, `movies-small.json`. You are also provided with an HTML file `movies.html` that contains a web page that can be used as a basis for your program. It links to a file `movies.js`, which is currently empty, where you should add your JavaScript code.

The specific requirements for the program are:

1. The JSON data about the movies should be read in from the file provided.

2. The data that is read in should be used to populate the table with the id `movieTable` inside the HTML file. It has four columns; year, title, genres, and cast. These correspond to the property names of the objects in the JSON file. There should be a row in the table for each object in the JSON file.

3. It should be possible for the user to select a year from the select element with the id `selectYear` (inside the HTML file). When they do this the table should update to show only the films for that

year. The select element should also allow the user to select an option labelled "All" that updates that table to show the films for all years.

4. It should be possible for the user to select by genre for the films (shown in the corresponding field in the JSON file) using the select element with the id `selectGenre` (inside the HTML file). They should be able to select only from genres that are included in the file. Note that films can have multiple genres, and some are empty, so this should be dealt with appropriately. The select element should also allow the user to select an option labelled "All" that updates that table to show films in all genres. This filter should combine appropriately with the year filter.

Feel free to edit any of the provided files. If you do so, remember to mention it in your report.

There are several possible additional pieces of functionality you could add to your code. For example:

- the JSON contains cast data, for which you could add a search option.
- You could add additional information to the JSON data and allow the user to filter based on that.
- You could add links to the Wikipedia or IMDB entry for each film, where it exists

These are not formal requirements; instead they (or similar additions) demonstrate that your code can form the basis for useful improvements. To achieve a grade higher than 17, you will need to include some extension elements.

Report

Your report should include the output of your programs in snapshot and a description of your design choices and any difficulties that you have encountered. You should show evidence of testing your programs. The report should not exceed 1500 words (advisory limit) in length but may have any number of diagrams.

Deliverables

Code: save your program as a pair of an HTML and JS file, where the HTML links to the JavaScript file `movies.js`.

Save your report as a PDF document. Now package up your code and PDF report into one `.zip` file (for eg, `cs5002-assignment4.zip`). You need to submit this zip file to the MMS in the slot for Assignment 4.

Marking

This practical will be marked according to the guidelines at https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html. Note that to get a grade higher than 17, you must include some extension

- Standard lateness penalties apply as outlined in the student handbook at https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html
- Guidelines for good academic practice are outlined in the student handbook at https://info.cs.st-andrews.ac.uk/student-handbook/academic/gap.html