Web Data with XML, JSON and AJAX

Final Marked Assignment (FMA)

Introduction

The Web Data with XML, JSON and Ajax FMA accounts for 75% of the module marks. It covers the bulk of the materials covered in sessions 5 – 8, notably JSON, JSON Schema, Ajax and Web APIs.

Outline

You will create two small-scale UK weather information applications. For the first application, you will start by creating a JSON data file capable of holding UK weather summary data. You will then use jQuery to extract data from your JSON data file, and to process and display the data in HTML format. For the second weather application, you will use jQuery to access and query current UK weather conditions from a weather data API. You will then display the resultant data in HTML format.

FMA Specifications

Task 1 – Create a Current Weather Application (50%)

JSON

Create a JSON file (*weather.json*) with a structure capable of holding current weather data for cities across the UK. The file should be capable of holding the following data:

city id
city name
current conditions
temperature
wind speed
wind direction
wind chill factor

JSON Schema

Create a JSON schema (*weatherschema.json*) against which weather data from *weather.json* can be validated. The schema should enforce *realistic* data types and restrictions for each data element.

Add dummy weather data for several UK cities to *weather.json*. The data you add should validate against *weatherschema.json*.

jQuery & HTML

Create a jQuery Ajax script (*weather.js*) that will extract data from *weather.json,* and display that data on a HTML page (*weather.html*) in table format. The data should automatically update at a realistic pre-set interval. This should be achieved using a *setTimeout* method. The outputted HTML should be formatted to professional standard using CSS.

Note that there should be NO DELAY in loading the data when the page is first opened.

Also, include a weather icon (provided) for each city weather report.

Sources of all code and code snippets must be cited in comments in the JS code. Failure to do so will result in the element being awarded zero.

Code Explanation

Copy your code to a Word document. Annotate each chunk of the code with a *full* explanation of what the code does. This includes variables, methods, structural constructs (e.g. conditionals), etc.

Example:

Iterates over items in an array, using an anonymous function to access both the current data ememement and its index.

```
$.each ( [ "a", "b", "c" ], function( i, element ) {
```

Concatenates the element at the current array index to '#Index' and writes the element to the console.

```
console.log ( "Index #" + i + ": " + element );
} );
```

Save your Word document as *task1-code-explanation.docx*.

Submission Procedures

Your finished application should contain the following files:

- *weather.html*
- *weather.js*
- *weather.json*
- *weatherschema.json*
- *styles.css*

You should publish your application (HTML, JavaScript, CSS and JSON files) at the following URL:

```
http://titan.dcs.bbk.ac.uk/~username/wd/fma/task1/weather.html
```

Zip your application files. Name the file *username_wd_fma_task1.zip*.

Task 2 – Create a Weather Application based on API Data (50%)

The API

To build the UK weather application for this part of the FMA, you will use the OpenWeatherMap API. To use this API you will first need to register and get yourself an API key. There are detailed instructions on the OpenWeatherMap API help page on how to work with the API (e.g. how to create a request URL).

The Application

The application should work as follows. Users should be able to choose a UK country (England, Scotland, Wales, or Northern Ireland) from a menu on a HTML page. Depending on which country the user chooses, corresponding city name data for the chosen country should be dynamically loaded into a second menu. The user should then be able to select a city name from the second menu. Upon selection, the weather for the selected city should be retrieved from the API, and displayed as HTML.

UK city data for the menus is provided in the FMA resources folder in Moodle.

The following weather data items should be displayed for a selected city:

- City name
- Date
- Weather conditions
- Temperature
- Wind speed
- Wind direction
- Weather icon.

The following display requirements should apply:

- Temperature must be displayed in *Celsius* and *fahrenheit*. If the temperature is above 35°C or below -5°C, a severe weather warning must be issued.
- Wind speed must be displayed in *miles-per-hour and kilometres per hour*. If the wind speed is greater than 50mph (80.47kph), a severe weather warning must be issued.
- Wind direction must be displayed as a degree (e.g. 90°) and as a textual description (e.g. Southerly, South Westerly, Westerly, etc.).
- The date must be displayed in standard date format (e.g. DD-MM-YYYY).

Presentation of the application must be to professional standard.

Sources of all code and code snippets must be cited in comments in the JS code. Failure to do so will result in the element being awarded zero.

Code Explanation

Copy your code to a Word document. Annotate each chunk of the code with a *full* explanation of what the code does. This includes variables, methods, structural constructs (e.g. conditionals), etc.

Save your Word document as *task2-code-explanation.docx*.

Submission Procedures

Your finished application should contain the following files:

- *apiweather.html*
- *apiweather.js*
- *styles.css*
- *england-cities.html*
- *scotland-cities.html*
- *wales-cities.html*
- *nireland-cities.html*

You should publish your application files (HTML, JavaScript, CSS) at the following URL:

http://titan.dcs.bbk.ac.uk/~username/wd/fma/task2/apiweather.html

Zip your application files. Name the file *username_wd_fma_task2.zip*.

Final Submission

You must submit a single .zip file in Turnitin. The file should be called:

username_wd_fma.zip

The zip file should contain the following:

- *task1-code-explanation.docx*
- *username_wd_fma_task1.zip*
- *task2-code-explanation.docx*
- *username_wd_fma_task2.zip*

Completing the FMA

The FMA must be completed and submitted electronically in the assignment dropbox in *Moodle* BEFORE the FMA deadline for your class.

Begin your work early, as the FMA is a substantial task that requires planning and effort to complete satisfactorily.

Getting support

Support for the FMA work will be available from your tutor until two weeks before the assignment deadline.

Getting feedback

The FMA will be marked by your tutor and then second marked by another tutor. This process can

take up to eight weeks. Once all the required marking and second marking has been completed, your grade and your feedback will be uploaded to Moodle.

Backing up files

Always keep a back-up copy of all work submitted for assessment in case of unforeseen submission problems.

Plagiarism

Plagiarism, which is claiming the work of others as your own, is a serious offence and can result in your exclusion from all colleges of the University of London. You should be aware that we use a range of automated tools to spot potential plagiarism in spreadsheets, databases, programme code and text documents. Providing you clearly reference work done by others that you have included in your FMA you will not be penalised.

In the course of completing the assignment, we acknowledge that you will research code from books and from online sources. *Ideas* and *techniques* from these sources may be used in the completion of your own work. HOWEVER, your own work MUST differ significantly from any third-party sources. If it does not, this will constitute plagiarism. You must also clearly reference any third-party sources you have used.

Likewise, we acknowledge that some students will work together collaboratively to solve problems. Again, if you do this, each student's final submission must be markedly different. If your work is not markedly different from another student's work, again, this will consitute plagiarism.