# MET CS601: Module 2 Assignment

## General Rules for Homework Assignments

- You are strongly encouraged to add comments to your source code. Doing so will help your facilitator to understand your logic/approach and grade your work more accurately.
- You must work on your assignments individually. You are ***not allowed*** to copy the answers from others. However, you are encouraged to discuss approaches to the homework assignment with your facilitator
- You are expected to write your own code for all assignments. You may use an IDE or advanced text editor for your assignments, but you *must* ***not*** use any auto generated code provided by such tools or other applications. So be sure to write your own code in the editor window, don't use the WYSIWYG builder (if applicable).
- Do not use any unapproved code libraries or frameworks.
- Each assignment has a strict deadline. However, you are still allowed to submit your assignment within ***two (2) days*** after the deadline with a penalty. 15% of the credit will be deducted unless you made previous arrangements with your facilitator. Assignments submitted 2 days after the deadline will not be graded.
- When the term *lastName* is referenced in an assignment's file or folder name, please replace it with ***your*** last name.

Create a new folder/directory named **CS601_HW2_*lastName***. Place your solution(s) to the assignment requirements in this folder.

| *NOTE: THIS DOCUMENT CONTAINS MULTIPLE PAGES* |
|---|

# Asynchronous Web App Development

This assignment tasks you with creating a dynamic web application that manages and displays custom JSON data. You will design your own JSON structure, save this data to a local server environment or upload it to a web server, and then fetch it for use in your application. The interface will feature a drag-and-drop functionality allowing users to interact with the displayed data elements. The aim is to develop a cohesive application that combines data creation, fetching, dynamic updating, and interactive user interfaces, emphasizing asynchronous web operations and client-side data manipulation.

## Requirements

Complete the following using only HTML5, CSS3, and JavaScript (no other languages, libraries, or frameworks).

A. Design JSON Structure
   1. Create a JSON file that includes two distinct sets of items, such as "fruits" and "vegetables". Each set should have at least 5 items with at least the attributes **id**, **name**, and **category**.
   2. Use a JSON validator, such as http://jsonlint.com (plenty available online), to check your file for correct formatting and eliminate any errors.
   3. Choose between setting up a local server (e.g., using Node.js and Express) to serve your JSON file, or uploading the JSON file to a web hosting service where it can be accessed via HTTP requests. Your instructor or facilitator can assist you with this.

B. HTML Structure
   1. Design the basic HTML layout for displaying your data.
   2. Create an empty container element, such as a **div** or **ul**, where the parsed JSON data will be dynamically inserted.
   3. Create two separate drop zones for the two categories of data, clearly labeled with the category names (e.g. Fruits and Vegetables).

C. Fetch Your JSON Data and Utilize the HTML5 Drag-and-Drop API
   1. Write JavaScript to asynchronously fetch the categorized data from the JSON file using the Fetch API when the application initializes.
   2. Create or use high-order functions to loop through the data and generate a list of mixed items to display it to the DOM.
   3. For each of your item
      a. Set it as a draggable element and tag each element with its category (e.g. “fruit” or “vegetable”), which will determine the allowed drop zone.
      b. Register the **dragstart** event to fire when the user starts dragging the element.
   4. For each drop zone (target)
      a. Register a **dragover** event to fire when a dragged element is over this drop target.
      b. Register a **drop** event to fire when the user releases the dragged element over this drop target.

D. Run your program and verify that it works as outlined above.

## Assessment/Grading

Your assignment submission will be scored by the following criteria:

1. Strict adherence to the requirements stated above: 70%
2. Code validates without errors (warnings are OK): 10%
3. Overall quality of work and effort as determined by your facilitator: 20%
    a. Includes your project/assignment also have a README.md file

Thoroughly test the application to ensure the interface updates in real-time as items are dragged and dropped.

It is important that your code passes validation; you should use http://validator.w3.org for assistance.

You must also validate your CSS code as well, that can be done here: http://jigsaw.w3.org/css-validator/

## Submission

Export your **CS601_HW2_*lastName*** folder containing all the relevant sub-folders and files as a zip file and upload the zip file to the appropriate assignment submission area.