Web Fundamentals test

Objective

To assess what you’ve learned during the course.

Overview

In this test you’ll write code to practice using the web programming concepts which you’ve learned during the course. There are no detailed step by step instructions but you’ll find sufficient instructions for each task.   
  
There is no pass/fail mark for this assessment; however, your work will be uploaded to Canvas as a record of the progress you have made on the course.

You will have 2 hours to complete this task.  
  
a) Examine the Starter project

1. Load the Starter project for this test.
2. Run the Starter project. It will open a web browser (don’t worry if the web browser shows a “Not Authorised” message). In your web browser, navigate to the following URLs (where xxxx is the number which Visual Studio places into your browser’s URL bar when the project runs):
   1. <http://localhost:xxxx/OrdersService.asmx/getOrders>
   2. <http://localhost:xxxx/OrdersService.asmx/getOrderDetails?orderId=10248>
3. Examine the output from the web service, which is in JSON format, and make sure you understand what it is doing. Understand that getOrders returns an array of orders, and getOrderDetails returns an array of items in the order.

b) Build an Application

1. You will now build a web application which will:
   * 1. Show the user a list of orders
     2. When the user chooses an order, it will show the items in the order
2. You may choose whichever web technology you like from this course – JavaScript (not recommended), JQuery or AngularJS.
3. You should decide what your web application will look like

Hints:

1. Start by displaying the orders. This will be similar to exercises we’ve done in the Ajax chapters of the course. You should decide which fields to display to the user.
2. Then, build an event handler that detects when a user chooses an order. This will be similar to what we did in the Events chapters of the course.
3. Now, make it so that the event handler loads and displays the order details. Again, this will be similar to the Ajax exercises on the course, except that:
   1. You will need to pass a parameter (the chosen order id) to the web service
   2. You will need to decide how the web page will look when the user chooses an order. Will you show the orders at the top of the screen, and the order details on the bottom? Will clicking on an order result in going to a new page, or a new AngularJS route? Whatever method you choose, how will it find out the order number that was clicked on?

**Hints for AngularJS:**

To pass parameters into your Ajax call:

$http({

method: "GET",

url: "OrdersService.asmx/getOrderDetails",

**params: { orderId: myOrderId }**

}).then(callback function);

**Hints for JQuery:**

To remove all rows from a table:

$("#myTable tr").remove();

1. Once you have a working project, use HTML and CSS to improve the appearance of your project.
2. If you have time to spare, add extra functionality to make your app have a more professional feel. Examples might be: searching and sorting, some “loading” text while the data is loading, or some error handling in case the service returns an error.

\*\* End

Marking

|  |  |  |
| --- | --- | --- |
| Task | Value | Grade achieved |
| HTML works correctly | 2 |  |
| CSS used appropriately | 4 |  |
| Orders displayed correctly | 10 |  |
| User can choose an order | 12 |  |
| Order details displayed correctly | 12 |  |
| Additional functionality | 10 |  |
| Total | **50** |  |