SER421 Fall 2016 Online Lab6: HTTP, AJAX, and APIs  
Due 11/25/16 at 11:59:00pm via online submission to Blackboard

**Objectives:**

For this lab you will develop an HTML5/Javascript application using AJAX against a REST API. You may complete the Activity by yourself or with ONE partner *with whom you have not partnered with before*.

**Activity: Use AJAX to access a REST-like API**

*www.apixu.com* provides a Web API that returns JSON weather data. You can sign up for free access for 5K API calls per month. Once you do you will get example URLs to the current weather and forecast endpoints with your own API key. You will want to save the endpoints and the API key. They also have a nice "Interactive" tab for experimenting with your own API calls.

Write a complete web application that does the following:

1. (30) Displays 2 cities and their associated data. The data should be retrieved and parsed out of the JSON at URLs like the above via an AJAX call. The data you should display in a table (2 rows):
   1. The first city should be "Phoenix"
   2. A second city name custom to your application. (Paris, London, you pick)
   3. For each row, have a column for the city name followed by:
      1. A timestamp when the data was last updated.
      2. Temperature in Fahrenheit, plus "feels like" temperature.
      3. Humidity – a percentage. Example: 70 means “70% humidity”
      4. Wind speed in miles per hour
      5. Conditions - text
2. (15) Populate a 3rd city row by selecting from a set of 5 cities in a dropdown. You may populate the dropdown with any 5 cities you like. The website has a list of cities supported. When a new city is selected, you should populate its data in a 3rd row.
3. (15) At the top of the page, display the following lines:
   1. “The average temperature is AAA and the hottest city is TTT”
   2. “The city with the nicest weather is XXX”. This should be custom logic you create with the constraint that your computation involve all of the cities and all of the values from 1.c.ii to 1.c.iv above.
4. (20) Next to each city name put a "forecast" button that, when clicked, puts text below the table describing the weather forecast for tomorrow (see the Examples) *for that city* using the values for "day" and "night" (see the JSON).
5. (20) Provide a "Refresh" button at the bottom of the table that causes the data values *for the entire table* to be updated as well as recompute requirement #3 based on the new values.

**Extra Credit: (50 points)**

*Task 1 (15 points):* When selecting "Refresh" in requirement 6, add a visual indicator to indicate whether a value is going up or down for the values in 1.c.ii to 1.c.iv. That is, indicate (using color, text, icons – your choice) whether it is getting warmer/colder/same, more/les/same humid, and more/less/same windy. Indicate the delta in time (1.c.i) as a hard number.

*Task 2 (35 points):* Find a client-side charting library on the web. Use it to chart the forecast changes in temperature for a city hourly for tomorrow. Augment the "forecast" feature in Requirement 5 to display this chart below the forecast text.

**Hints:**

* This lab may appear somewhat daunting at first, but spend 15 minutes on the API website with their tools and by inspecting JSON in the browser and you will see it is not that difficult.
* I do expect you to use an HTML table to render the content for requirements 1-3.
* I put the requirements in an order I suggest you use to approach the development of your solution. Get a requirement stable before moving on to the next one, and save your work along the way!
* Be sure to check for non-200 responses and have basic error-checking – your app should not crash or ruin the rendering of the display table if an API AJAX call is not successful; instead it should show an appropriate error message!
* Timestamps in epoch is measured in milliseconds from 1/1/1970 (when you see timestamp\_epoch in the JSON).

**Submission:**

Submit via a zipfile to Blackboard by the due date. Name the file <asurite>[\_<asurite2>]\_lab6.zip. You may add a README.txt in the zipfile if there is anything you want us to take into consideration. If you did the Task with a partner please indicate so in a header comment of your task javascript file. There will be no extensions for this lab as it is due to the end of the session! Note the modified submission dates and late deadlines due to the Thanksgiving holiday. Also note this lab is the last chance to use your free late pass as you may not use it on the last lab! Don't forget your Readme.txt!