**COMM644  
Assignment 10**

**Part 1 - Using the jQuery .load() Shorthand Method**

In this part of the assignment you will be working with the jQuery .load() shorthand method. In the lectures for this topic you saw that you can programmatically load HTML content from another HTML document by simply clicking a link within the web page. You will be doing more of the same thing here.

* Open index.html.
* Within the <script> tag, replace the existing comment with a document ready function.
* Write the necessary event handlers to handle the click event for each of the 3 links within the index.html page.
* When each link is clicked, the corresponding section within the solutions.html page should be loaded into the div tag uniquely identified as solution.

**Part 2 - Using the jQuery .get() Shorthand Method**

In this part of the assignment you will be working with the jQuery .get() shorthand method. In the lectures for this topic you saw that you can programmatically load XML data into a web page using a server-side script such as a PHP file as an intermediary. Here you will connect directly to an XML file and retrieve its contents.

* Open index.html.
* Within the <script> tag, replace the existing comment with a document ready function.
* Use the $.get() method to point to the team.xml file. Upon successful load, it should call a function to execute some code. This function can be a callback function or a user-defined function…it’s up to you. Don’t forget to pass in a parameter within the function…this will represent the actual XML data and will be used to iterate through.
* Within the function, use method chaining to iterate through each of the child nodes in the management node. You may have to go back to the first lecture to figure this one out!
* Within the .each() method, append the contents of the name, title, and bio nodes to the div element uniquely identified as team. Don’t forget to concatenate some simple HTML markup to help format the text on the web page.

**Part 3 - Using the jQuery .getJSON() Shorthand Method**

In this part of the assignment you will be working with the jQuery .getJSON() shorthand method. In the lectures for this topic you saw that you can programmatically load JSON data into a web page using the .getJSON() method. Here you will recreate the code outlined in this lecture to make the code work for you.

* Create your own JSON file. Make sure to create a unique object name and populate the object with at least 5 items. Each item should have at least 2 sets of key/value pairs.
* Open index.html.
* Within the <script> tag, replace the existing comment with a document ready function.
* Use the $.getJSON() method to point to the JSON file that you just created.
* Use a set of nested .each() methods to iterate through the object and then the key/value pairs too. Display the values of each pair within the div element uniquely identified as result.

**Part 4 - Using the jQuery .ajax() Shorthand Method**

In this part of the assignment you will be working with the jQuery .ajax() shorthand method. In the lectures for this topic you saw that you can programmatically load XML or JSON data into a web page using the .ajax() shorthand method. Here you will rewrite the code that you wrote in part 3, using the .ajax() shorthand method.

* Rewrite the code that you wrote in lab 3 so that you are using the .ajax() shorthand method instead of the .getJSON() method.
* You must include and set the following options:
  + type
  + url
  + timeout
  + datatype
  + beforeSend: display a loading message on the page
  + error: display a jQuery UI dialog box with an error message inside
  + success: this is where you’ll process the data returned