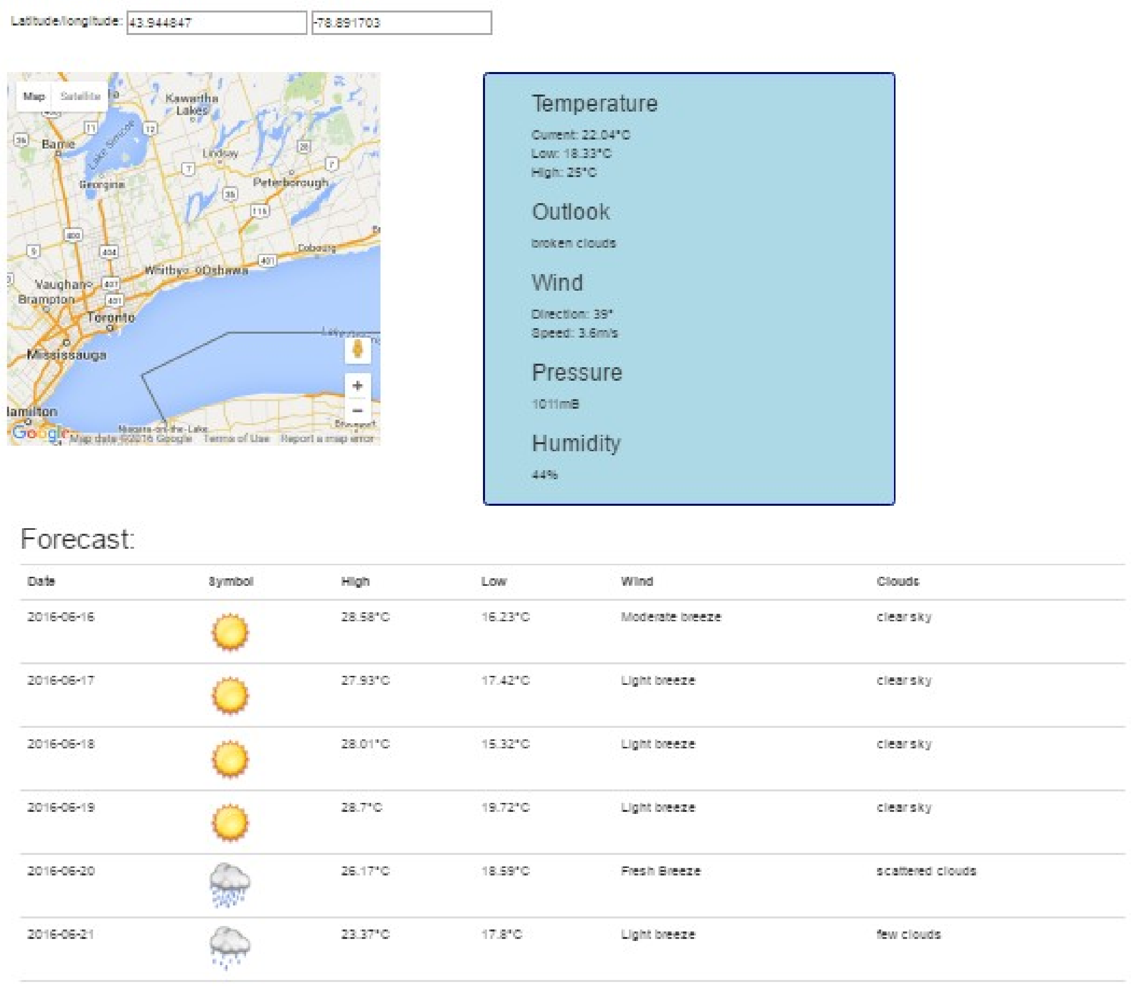
Faculty of Science

Course:	CSCI 3230U: Web Application Development
Assignment:	#2
Topic:	jQuery and web services

***Note**: This assignment is supposed to be submitted in teams of two. However, individual submissions are acceptable as well.*

Overview

This assignment collects information from two different sources (using the same web source, but different API endpoints) to produce its output. One source provides current weather data for a particular location (in JSON format), and the other provides a forecast for that location (in XML format). The page will also show a Google Map of that same location.



Latitude/longitude: 43.944847 -78.891703

Temperature

Current: 22.04°C
Low: 18.33°C
High: 25°C

Outlook

broken clouds

Wind

Direction: 39°
Speed: 3.6m/s

Pressure

1011mB

Humidity

44%

Forecast:

Date	Symbol	High	Low	Wind	Clouds
2016-06-16	☀	28.58°C	16.23°C	Moderate breeze	clear sky
2016-06-17	☀	27.93°C	17.42°C	Light breeze	clear sky
2016-06-18	☀	28.01°C	15.32°C	Light breeze	clear sky
2016-06-19	☀	28.7°C	19.72°C	Light breeze	clear sky
2016-06-20	🌧	26.17°C	18.59°C	Fresh Breeze	scattered clouds
2016-06-21	🌧	23.37°C	17.8°C	Light breeze	few clouds

Figure 1 – The completed page

The Main Page

The main page of your application will be relatively simple. It will contain only two text fields and a button ("Go"). When the user clicks this button, your JavaScript code will issue two AJAX requests, using jQuery, and will show the Google map.

Code Listing 1 – The HTML for the main page
```
<!DOCTYPE html>
<html>
  <head>
    <title>Consuming Web Services</title>
    <script type="text/javascript" src="http://code.jquery.com/jquery-
latest.min.js"></script>
    <script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=<YOUR GOOGLE API KEY GOES
HERE>"></script>
    <script type="text/javascript" src="asmt2.js"></script>
    <link rel="stylesheet" href="asmt2.css" />
  </head>
  <body>
      <div id="form">
        Latitude/longitude:
        <input type="text" name="lat" id="lat" value="43.944847" />
        <input type="text" name="lon" id="lon" value="-78.891703" />
        <button id="goButton">Go</button>
      </div>

      <div class="container-fluid">
        <div id="map-canvas" class="col-sm-6"></div>
        <div id="weather" class="col-sm-4 col-sm-offset-1"></div>
        <div id="forecast" class="col-sm-11"></div>
      </div>
  </body>
</html>
```

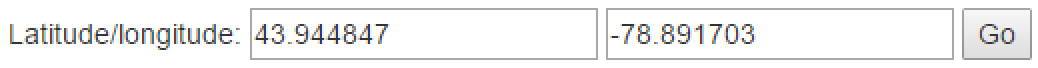


Figure 2 – The page before submission

Sign-up for API Keys

Both the weather API and Google Maps' API require that you sign up for API keys. The weather API that we will be using is called Open Weather API. You can use a throwaway E-Mail for this account, if you like. You can create an account, and an API key, at the following URL:

- https://home.openweathermap.org/users/sign_up

Google Maps requires a Gmail login. It is very likely that you will be using a Google API at some time in the future, since they are so ubiquitous, so this shouldn't be too much to ask. You can obtain a key for the JavaScript Google Maps API at the following URL:

- https://developers.google.com/maps/documentation/javascript/get-api-key

Button Event Handler

Using jQuery, register an event handler for the "Go" button. This event handler will extract the latitude and longitude values from the two text fields, and then will execute three functions, each of which will take the latitude and longitude as arguments:

- downloadWeather
- downloadForecast
- showMap

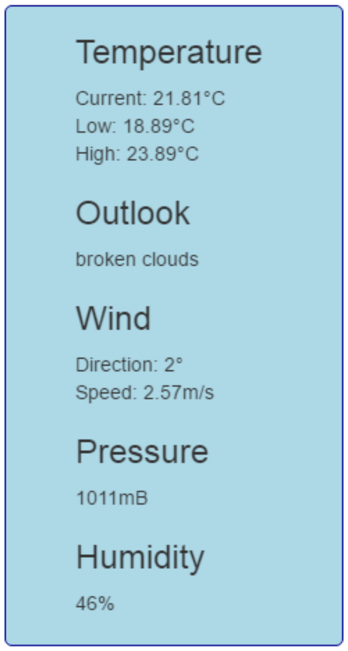
Populating the Weather Box

The `downloadWeather()` function will issue a jQuery-based AJAX request, sent to Open Weather Map's weather service. This service API can be found at the following URL:

- `http://api.openweathermap.org/data/2.5/weather?lat=43.944847&lon=-78.891703&units=metric&APPID=<YOUR KEY GOES HERE>`

***Note**: Try this by visiting this URL in your browser before you try accessing it through JavaScript.*

The function will extract data from the JSON response and put it into the weather div. A typical JSON response for one of these requests is given in code listing 2. The output should resemble figure 3.


Temperature

Current: 21.81°C
Low: 18.89°C
High: 23.89°C

Outlook

broken clouds

Wind

Direction: 2°
Speed: 2.57m/s

Pressure

1011mB

Humidity

46%

Figure 3 – Contents of the weather DIV with web service data

Code Listing 2 – Example JSON response for the weather API

```
{"coord":{"lon":-78.85,
          "lat":43.9},
 "weather":[{"id":803,
             "main":"Clouds",
             "description":"broken clouds",
             "icon":"04d"}],
 "base":"cmc stations",
 "main":{"temp":21.81,
         "pressure":1011,
         "humidity":46,
         "temp_min":18.89,
         "temp_max":23.89},
 "wind":{"speed":2.57,
         "deg":2,
         "gust":5.14},
 "clouds":{"all":68},
 "dt":1466084055,
 "sys":{"type":3,
        "id":39825,
        "message":0.0071,
        "country":"CA",
        "sunrise":1466069543,
        "sunset":1466125226},
 "id":6094578,
 "name":"Oshawa",
 "cod":200}
```

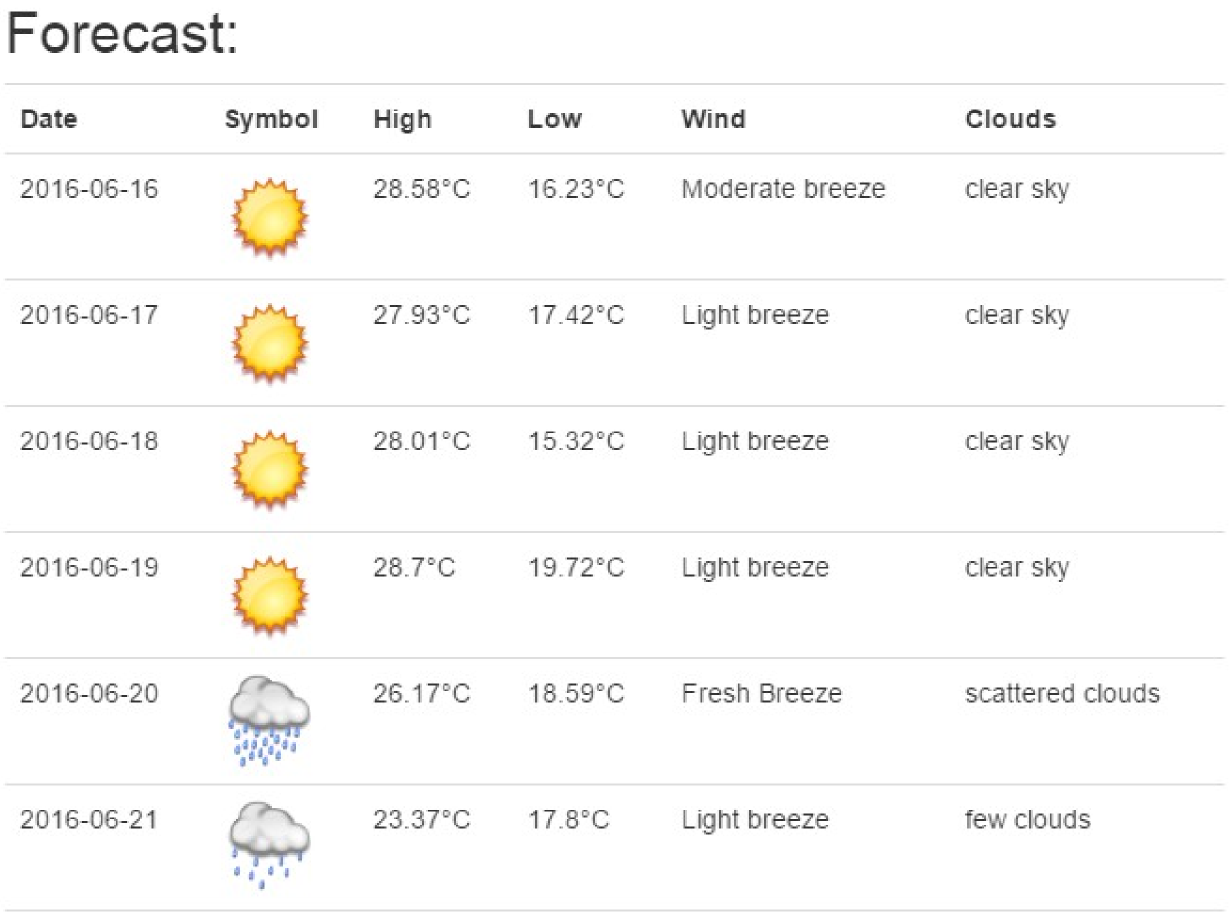
Populating the Forecast Table

The `downloadForecast()` function will issue a jQuery-based AJAX request, sent to Open Weather Map's forecast service. This service API can be found at the following URL:

- `http://api.openweathermap.org/data/2.5/forecast/daily?cnt=10&mode=xml&lat=43.944847&lon=-78.891703&units=metric&APPID=<YOUR KEY GOES HERE>`

***Note**: Try this by visiting this URL in your browser before you try accessing it through JavaScript.*

The function will extract data from the XML response and put it into the weather div. A typical XML response for one of these requests is given in code listing 3. The output should resemble figure 4. The images can be found on the Assignments page, and are named after the 'number' attribute of the 'symbol' tag in each 'time' (e.g. 800.png).



Forecast:

Date	Symbol	High	Low	Wind	Clouds
2016-06-16	☀	28.58°C	16.23°C	Moderate breeze	clear sky
2016-06-17	☀	27.93°C	17.42°C	Light breeze	clear sky
2016-06-18	☀	28.01°C	15.32°C	Light breeze	clear sky
2016-06-19	☀	28.7°C	19.72°C	Light breeze	clear sky
2016-06-20	🌧	26.17°C	18.59°C	Fresh Breeze	scattered clouds
2016-06-21	🌧	23.37°C	17.8°C	Light breeze	few clouds

Figure 4 – Contents of the forecast DIV with web service data

Code Listing 3 – Example XML response for the forecast API

```
<?xml version="1.0"?>
<weatherdata>
    <location>
        <name>Oshawa</name>
        <type/>
        <country>CA</country>
        <timezone/>
        <location altitude="0" latitude="43.90012" longitude="-78.849571"
geobase="geonames" geobaseid="6094578"/>
    </location>
    <credit/>
    <sun rise="2016-06-16T09:32:23" set="2016-06-17T01:00:26"/>
    <forecast>
        <time day="2016-06-16">
            <symbol number="800" name="clear sky" var="02d"/>
            <precipitation/>
            <windDirection deg="97" code="E" name="East"/>
            <windSpeed mps="5.53" name="Moderate breeze"/>
            <temperature day="28.44" min="16.23" max="28.58" night="16.23"
eve="25.83" morn="21.96"/>
            <pressure unit="hPa" value="991.04"/>
            <humidity value="50" unit="%"/>
            <clouds value="clear sky" all="8" unit="%"/>
        </time>
        <time day="2016-06-17">
            <symbol number="800" name="clear sky" var="01d"/>
            <precipitation/>
            <windDirection deg="73" code="ENE" name="East-northeast"/>
            <windSpeed mps="2.21" name="Light breeze"/>
            <temperature day="27.58" min="17.42" max="27.93" night="17.42"
eve="25.62" morn="21.26"/>
            <pressure unit="hPa" value="999.15"/>
            <humidity value="51" unit="%"/>
            <clouds value="clear sky" all="0" unit="%"/>
        </time>
        ...
    </forecast>
</weatherdata>
```

Showing the Map

For this part of the assignment, you will use the Google Maps API to display a map of the user's location. Use the div with id 'map-canvas' for this purpose. The code for the `showMap()` function will be based off of the code found in the following tutorial:

- https://developers.google.com/maps/documentation/javascript/examples/map-simple.

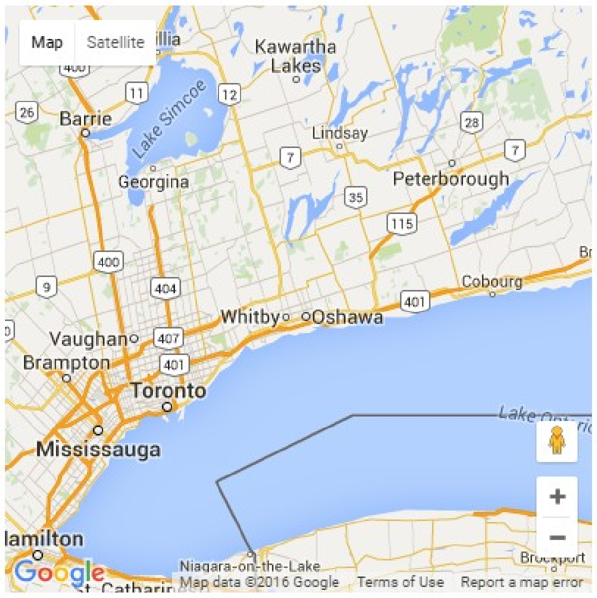


Figure 5 – The embedded Google Map in the map-canvas DIV

How to Submit

Add your HTML, CSS, and JavaScript files to a ZIP file (e.g. RandyFortier_100539147_Assignment2.zip) and submit to the Assignment 2 drop box on Blackboard.