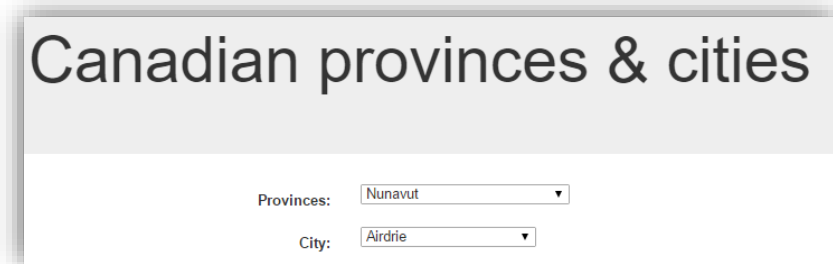


You are expected to complete this exercise during your lab session, show the lab instructor your work, and submit it into the D2L lab site for COMP1536. If you do not finish, then submit whatever you have by 11:00 PM on the day of your lab for late marking. This will, however, cost you a deduction of 10% of the maximum mark.

AJAX

Build PHP, JavaScript, jQuery code that makes AJAX requests to a server-side web application and retrieves cities in a Canadian province from a text file.

- 1) Unzip a file named *ajax_start.zip*.
- 2) Rename the *ajax_start* folder to *LastName_Firstname_ajax*.
- 3) View the contents of the *province_city.txt* text file. It contains all the provinces and cities in Canada.
- 4) Open the *LastName_Firstname_ajax* folder in your favorite web IDE like *Brackets*, *Sublime*, *VS Code*, etc. You will see files *index1.php*, *province_city.txt*, *Canada.css* and *ajax-loader.gif* files.
- 5) Start your web server.
- 6) Open *index1.php* in the Brackets editor and launch it in your browser. You will see the following page.



By inspecting the PHP code in page *index1.php*, you will determine that data is being read from the text file named *province_city.txt*. This page displays only the cities found in the province of Alberta. Notice that changing the province does not result in any changes in the *City* drop-down-list.

Your mission:

Your mission is to *AJAX-ify* the application using both *Raw-AJAX* and *jQuery-AJAX* approaches such that when the user changes the province, the city drop-down-list gets populated without the page being refreshed.

Part 1: Raw-AJAX (*index1.php*)

- 1) In the *index1.php* file add the following event handler for the *Province* *<select>* tag:

```
onchange="getCitiesFromProvince()"
```

- 2) Add the following tag that points to a JavaScript file right before the closing *</head>* tag:

```
<script type="text/javascript" src="rawAjax.js"></script>
```

- 3) Create a PHP file named *cities_from_province.php*. This PHP page receives a province name through a query string on the URL line and echoes back the *cities* in that *province*. Cities will be separated by the '~' *delimiter*. Here is the code that you will place in the *cities_from_province.php* file:

```
<?php
# load all the lines in the text file into an array where each line is
# an element in the array
$myArray = file("province_city.txt");
sort($myArray);
$result = "";
if (isset($_GET["prov"])) {
    $province = $_GET["prov"];

    # Find cities pertaining to province selected by user
    foreach ($myArray as $provCity) {
        if (preg_match("/^$province/", chop($provCity))) {
            // remove newline character from end of line
            $provCity = chop($provCity);
            // make $province first string and $city second string
            list($province, $city) = explode("-", $provCity, 2);
            if ($result == "")
                $result .= $city;
            else
                $result .= "-" . $city;
        }
    }
}
# send result back to requesting client
echo $result;
?>
```

- 4) Test that *cities_from_province.php* works as expected by pointing to it in a browser as follows:

`http://localhost:8888/cities_from_province.php?prov=Nunavut`

Note: You may have to adjust the port number above depending on your web server's environment.

The above should result in the cities of *Nunavut* being displayed in your browser separated by the delimiter '~'. Once this works properly, then you are in a strong position to start on the next JavaScript phase.

- 5) Create a JavaScript file named *rawAjax.js* and add the following code to it:

```
var ajax = new XMLHttpRequest(); // we create the HTTP Object
```

The above instantiates the JavaScript *XMLHttpRequest* object that is fundamental to AJAX.

- 6) Next add the following JavaScript function to *rawAjax.js*:

```
function getCitiesFromProvince() {
    "use strict";
    var province, url;
    province = document.getElementById("provSelect").value;
    url = "cities_from_province.php?prov=" + window.encodeURI(province);
    ajax.open("GET", url, true);
    ajax.onreadystatechange = handleHttpResponse;
    ajax.send(null);
}
```

The above function reads the province selected by the user and makes a call to the PHP service at *cities_from_province.php* on the server. The response from the server is returned as a single string, which is then processed by a JavaScript function named *handleHttpResponse()*.

- 7) Here is some incomplete client-side JavaScript code pertaining to the *handleHttpResponse()* function referred to in item 6 above:

```
function handleHttpResponse() {
    "use strict";
    var cities;
    if (ajax.readyState == 4 && ajax.status == 200) {
        cities = ajax.responseText;

        // The cities string contains cities separated by the tilde (~).
        // Use JavaScript split() function to return all cities in an array.
        // Then, add code that populates cities drop-down list.
    }
}
```

You must complete the remaining code so that when the user changes *province*, the *cities* drop-down-list is populated appropriately with the *cities* in that *province*. Read the hints given at the very end of this document.

Part 2: AJAX with jQuery (index2.php)

- 1) Copy *index1.php* to file name *index2.php*.
- 2) Add the following links to a jQuery CDN in the appropriate place in your *index2.php* file:

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.8.3.js"></script>
```

- 3) Change the link to *RawAjax.js* so that it points to *jQueryAjax.js* instead. Ensure that the *<script>* tag for jQuery comes before the *<script>* tag for *QueryAjax.js*.
- 4) Delete the *onchange* event handler in the *select* tag with *id* of *provSelect*.
- 5) Create a JavaScript file named *jQueryAjax.js* and paste into it the following code:

```
$(function () {
    "use strict";
    $('#provSelect').change(function () {
        $('#animation').append('');
        var province = $('#provSelect').val();
        $.ajax({
            url: 'cities_from_province.php',
            type: 'GET',
            data: 'prov=' + province,
            success: function (result) {
                $('#citySelect')[0].options.length = 0;
                var index, cityArray;
                cityArray = result.split("~");
                for (index in cityArray) {
                    if ($.trim(cityArray[index]) !== "") {
                        $('#citySelect').append("<option/>", {
                            value: cityArray[index],
                            text: cityArray[index]
                        });
                    }
                }
                $('#loading').fadeOut(500, function () {
                    $(this).remove();
                });
            }
        });
    });
});
```

- 6) View *index2.php* in a browser. It should behave just like *index1.php*. This shows that you can accomplish the same task using either Raw-AJAX or the *\$.ajax()* jQuery function. Working with jQuery is much more elegant.
- 7) To see some animation while data is being retrieved, add the following <div> below the drop-down-lists in *index2.php*:

```
<div id="animation"></div>
```

Part 3

Return to *index1.php* and implement animation of the *images/ajax-loader.gif* using raw JavaScript. The animation is intended to give the user a visual indication that processing is happening while he/she waits. Therefore, animation should show while the request is being processed by the PHP service. Start the animation right after the *ajax.send(null)* statement and end it after data is successfully returned and injected into the cities dropdown list.

HINT:

- 1) Use the JavaScript *split()* function to split a string into an array of substrings.
- 2) The code to clear a drop-down-list named city is:

```
document.getElementById("city").options.length=0;
```

- 3) The code to add items to a drop-down-list named city is:

```
document.getElementById("city").options[ndx] = new Option("Burnaby");
```

- 4) Use the following code to change the CSS display property of an element:

```
document.getElementById("animation").style.display = "block";
```

- 5) Debug JavaScript in your browser with F12.

Submission

- Show your work to the lab instructor who will give you a mark on the spot.
- Zip all the files comprising your web site to filename *LastName_FirstName_set_week14.zip*. (Example: *Doe_Jane_1B_week14.zip*) and upload it to the *Week-14* dropbox in D2L.

Marking Guide

COMP1536 AJAX lab exercise marking guide:		
Name _____		Set _____
Task	Max Mark	Actual Mark
Pure JavaScript solution	5	
• Cities drop-down-list changes properly when province changes		
jQuery solution	3	
• Cities drop-down-list changes properly when province changes		
Animation works in Pure JavaScript solution	2	
TOTAL:	10	