HTTP5203 Coding Challenge

This coding challenge will use the toronto-parks.xml file provided. There are **three parts** to this challenge.

Part 1: XML Schema (15)

Create a Schema for the toronto-parks.xml file.

Link the XML file to the Schema file so it validates against it.

Part 2: XSLT (15)

Write the XSLT file needed to display the data for park location #9 as shown below. (Do **NOT** use position() for this. Select the Location based on LocationID element value.)

The bullet items show the FacilityDisplayName and FacilityName.

Make sure to link the XML file to the XSLT.

Suggested approach: Get the display working in general (e.g. for all parks), then worry about how to select by LocationID.

EGLINTON FLATS SPORTS FACILITY

Location: 3601 Eglinton Ave W

416 392-2486

Facilities

- Park Shelter - Pavillion
- Sport Field - Soccer Field 4 (P)
- Multipurpose Room - Clubhouse Room (C)
- Sport Field - Cricket Pitch 1 - Fergie Brown Park (P)
- Sport Field - Cricket Pitch 2 - Fergie Brown Park (P)
- Sport Field - Hockey Field 6 (P)
- Sport Field - Hockey Field 7 (P)
- Sport Field - Rugby Field 8 (P)
- Sport Field - Football Field 9 (P)
- Sport Field - Soccer Field 10 (P)
- Sport Field - Soccer Field 11 (P)
- Sport Field - Soccer Field 12 (P)
- Sport Field - Soccer Field 1 (P)
- Sport Field - Soccer Field 2 (P)
- Sport Field - Soccer Field 3 (P)
- Playground

Part 3: XML & JS (15)

Open the toronto-parks.xml file. This data was downloaded from the city of Toronto's open data catalogue and it contains a list of all the park facilities in Toronto.

Create an HTML page with a dropdown (HTML <select>) populated with park names. The park names (<LocationName>) should be the option text and the park ID (<LocationID>) should be the option value.

For example:

```
<select …>
  <option value="1">ASHBRIDGE'S BAY PARK</option>
  …
</select>
```

Do not hardcode it. Use Javascript to read from toronto-parks.xml and generate the select options by reading from the XML. (This is similar to reading and displaying the table in class but you're creating option elements instead of table rows.)

After making a selection and clicking on the button, use JS to display park details like below (remember to prevent the form from submitting if using a submit button). In the park details, you just need to print the LocationName, Address, and FacilityDisplayName (in a list format). To make it simpler, you don't need to account for FacilityName.

Suggested approach:

- You can use similar XPath from Part 2 to help you select park data on button click if your XPath selected the location by id number value. Rather than a hardcoded value of "9", use the id value from the select.
- XPath is not necessary to match a selected id against a LocationID (you can use a loop instead).
- The FacilityDisplayName <li> elements are generated with a loop.