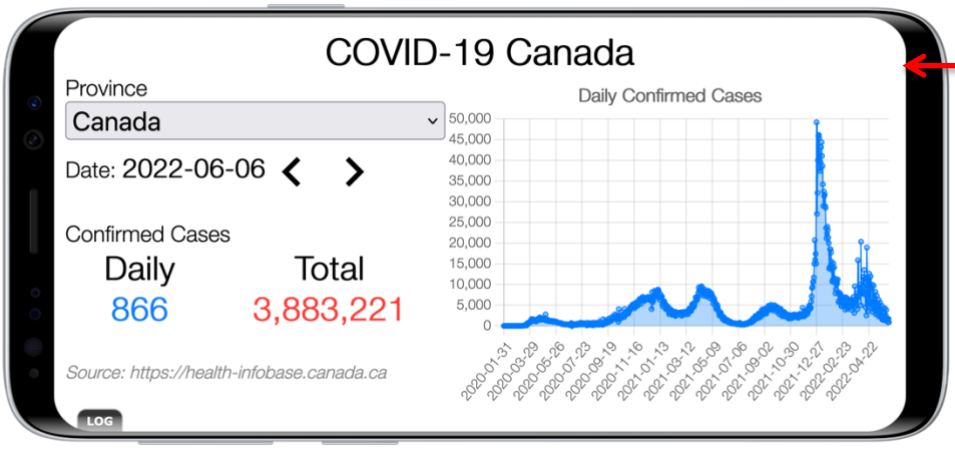
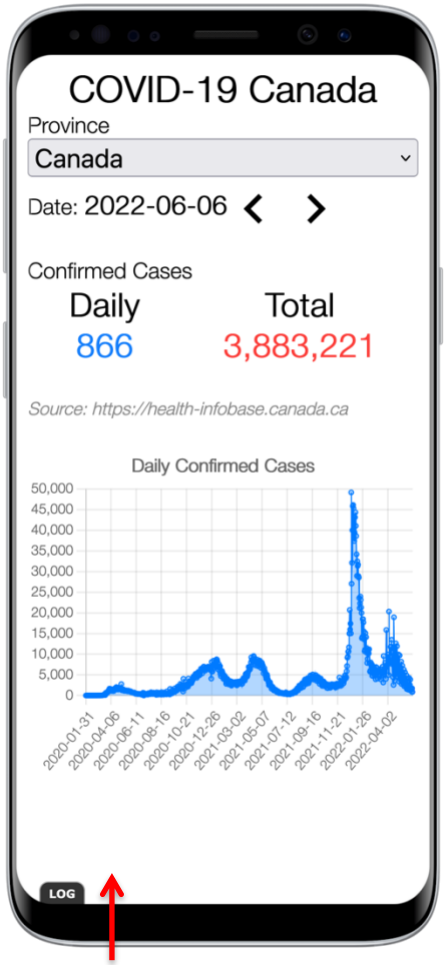
Assignment 2: COVID-19 Canada

Description

Write a responsive web application to report the number of COVID-19 confirmed cases in Canada and provinces since January 2020. When the HTML document is loaded, you request a web service to get the JSON data from a remote server asynchronously. Then, you display the Today's (the last date) daily and cumulative confirmed cases for the selected province. Also, you draw a line graph of the daily confirmed cases starting from the first date up to today.



Display Width > ~600 px (Landscape mode)



Display Width < ~600 px (Portrait mode)

The original source data is available from https://health-infobase.canada.ca, and it is converted to JSON format at http://ejd.songho.ca/ios/covid19.json. The JSON data structure is an array of dictionaries, and it looks like;

```
[
    {
        "pruid": 35,
        "prname": "Ontario",
        "date": "2020-01-31",
        "numconf": 3,
        "numdeaths": 0,
        "numtotal": 3,
        "numtests": 0,
        "numrecover": 0,
        "numtoday": 3,
        "numdeathstoday": 0,
        "numteststoday": 0,
        "numrecoveredtoday": 0,
        "numactive": 3
    },
    ...
]
```

NOTE: You need only 4 properties from the JSON data:

prname	: Name of the province
date	: Date of the data entered
numtotal	: Cumulative numbers up to date
numtoday	: Number of cases on the date

Requirements

- Must be responsive width, height, etc. using 2-column CSS flex layout
- Must be unobtrusive (no inline CSS and JavaScript)
- Must be validated (no errors or warnings in HTML, CSS & JavaScript)
- Can use native JavaScript, jQuery and/or third-party libraries for drawing charts
- Must load the JSON asynchronously using Promise + Arrow from http://ejd.songho.ca/ios/covid19.json
- Must include Canada (default), Ontario, Quebec and British Columbia into <select> options
- Must highlight the daily/total numbers with bigger font size and accent colours
- Must display the latest values in the daily/total numbers when the page is loaded
- Must Include the title and labels in the chart
- Must be responsive width/height for the chart (fit in the window)

Deliverables

An archive file, **Assignment2-*<yourname>*.zip**, which contains all the files (HTML, CSS, JavaScript).
***NOTE**: You must include the file header at the beginning of each file. The header must contain a short description, your name, email, date, etc.*

Submission and Due Date

Submit your deliverables to SLATE/Assignments/Assignment2 by **Saturday, Jul. 2, 11:59 PM**.
You may submit multiple versions, but only the latest version will be evaluated.
***NOTE**: Late submission will be deducted 10% per day. (max. 3 days)*
***NOTE**: Partial implementation will be accepted.*
***NOTE**: This assignment is individual work and subject to Sheridan Academic Integrity Policy.*

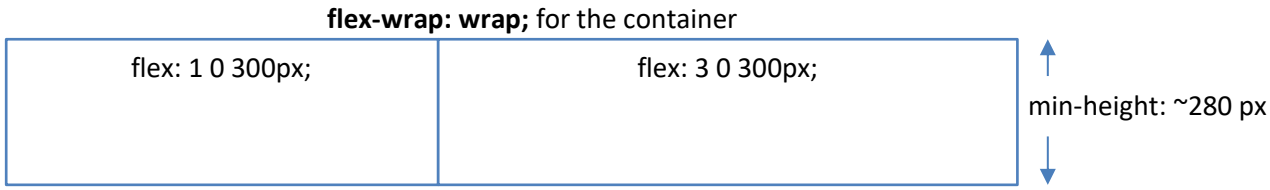
Tasks and Evaluations (Total 100 points)

Task 1: HTML (20 points)

1. Construct static DOM elements and placeholders in HTML page
2. Must have the title <h1> of the page, COVID-19 Canada
3. Load necessary CSS and JavaScript files
4. Not allowed inline CSS styles and JavaScript codes
5. Must contain “Canada” (*default selected*), “Ontario”, “Quebec” and “British Columbia” <option> tags in the province <select> element
6. Previous and next buttons to change the date

Task 2: CSS (20 points)

1. Use 2-column flex layout for the main block, allow automatically to wrap the columns for the width



2. The first column item contains province names in <select>, the latest date and the numbers for the daily/total confirmed cases
3. The second column item contains the chart, must be responsive (fit the chart in this block)
4. The daily and total numbers are aligned center
5. Use bigger font sizes for province names, date, daily and total numbers
6. Use accent colours for daily and total numbers

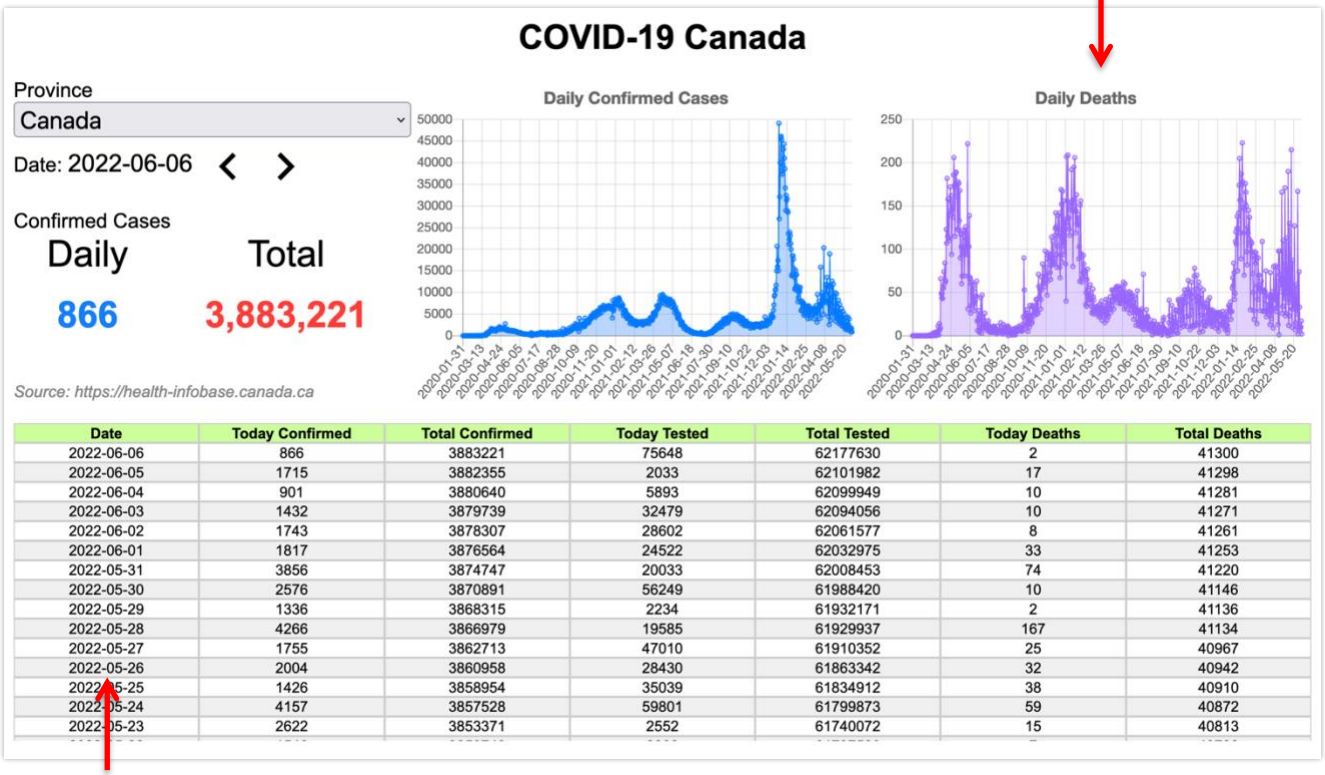
Task 3: JavaScript (60 points)

1. Load covid19.json from http://ejd.songho.ca/ios/covid19.json only once when the document is loaded
2. Must use Promise and Arrow function to load the JSON asynchronously
3. After the JSON loaded, create an array "dates" to store all the dates in ascending order from the beginning of COVID-19 data were collected to up to today. The date format is "yyyy-MM-dd" as string
   NOTE: The original JSON data contains only when the data are entered and skips if there is no records. However, this array must be continuous from the first date to up to date without hopping in order to draw the chart accurately.
   HINT: See **Q1** below to calculate the number of the array elements
   HINT: See **Q2** below to how to populate "dates" array
4. When user selects a province, generate another array "provinceData" to store only the selected provincial data from the original JSON data
   HINT: Use **filter()** function (See **Q3** below)
5. When user selects a province, you also create "values" array containing the daily confirmed cases for the corresponding selected province
   HINT: See **Q4** below
6. Display the daily and total confirmed cases from the latest data by default
7. Add "click" event handlers for the previous and next buttons to change the current date and to display the corresponding the daily and total numbers.
8. Draw the line graph corresponding the selected province by passing "dates" and "values" arrays
   HINT: Use 3rd-party chart library, such as Chart.js. An example of Chart.js, **test_chart.html** is available in SLATE/Assignment2 folder.

Bonus (20 points max)

- Add additional provinces into <select> (1 point per province, max 5 points)
- Add an additional graph with 3-column flex layout, e.g.: chart for the total confirmed cases (5 points)
- Add a table to display all data for the selected province (5 points)
- Add additional features to enhance the application (5 points)
  Example: http://ejd.songho.ca/ios/covid19_bonus.html

Line graph for daily death cases



Date	Today Confirmed	Total Confirmed	Today Tested	Total Tested	Today Deaths	Total Deaths
2022-06-06	866	3883221	75648	62177630	2	41300
2022-06-05	1715	3882355	2033	62101982	17	41298
2022-06-04	901	3880640	5893	62099949	10	41281
2022-06-03	1432	3879739	32479	62094056	10	41271
2022-06-02	1743	3878307	28602	62061577	8	41261
2022-06-01	1817	3876564	24522	62032975	33	41253
2022-05-31	3856	3874747	20033	62008453	74	41220
2022-05-30	2576	3870891	56249	61988420	10	41146
2022-05-29	1336	3868315	2234	61932171	2	41136
2022-05-28	4266	3866979	19585	61929937	167	41134
2022-05-27	1755	3862713	47010	61910352	25	40967
2022-05-26	2004	3860958	28430	61863342	32	40942
2022-05-25	1426	3858954	35039	61834912	38	40910
2022-05-24	4157	3857528	59801	61799873	59	40872
2022-05-23	2622	3853371	2552	61740072	15	40813

Display all the data into a table for the selected province

Q & A (Extra Notes)

Q1. How to calculate the number of elements for "dates" array?

The original JSON data does not store all data since the beginning of COVID-19. If there are no confirmed cases in a particular day, it omits the data in the JSON file. However, the "dates" arrays should be continuous from the first date to the last date without skipping/hopping in order to draw an accurate chart. Use the difference of the milliseconds between 2 dates, then divide it by the ms per day.

```
const MS_PER_DAY = 24 * 60 * 60 * 1000;                              // ms a day
let firstTime = new Date(go.json[0].date).getTime();                 // first date in ms
let lastTime = new Date(go.json[go.json.length-1].date).getTime(); // last date in ms
let dateCount = (lastTime - firstTime) / MS_PER_DAY + 1;             // # of days

// example of counting dates from 2 arbitrary dates
let time1 = new Date("2020-01-31").getTime();
let time2 = new Date("2020-02-08").getTime();
let dateCount = (time2 - time1) / MS_PER_DAY + 1;    // should be 9
```

Q2. How to populate "dates" array?

Use a for-loop and compute the current date string from the index of the array using Date.**toISOString()** function.

```
// firstTime and lastTime are ms of the first date and last date
dates = []; // create/init an empty array
for(let i = firstTime; i <= lastTime; i += MS_PER_DAY)
{
    let date = new Date(i).toISOString().substring(0,10); // yyyy-MM-dd
    dates.push(date);
}
```

Q3. How to find the data for the selected province and create an array?

You may use your own for-loop, find the date where "**prname**" equals to the selected province and append it to array (Imperative programming). However, you can also use Array.**filter**() function instead (Declarative programming).
Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

```
go.json = ...;  // suppose go.json stores the original JSON data
go.provinceData = go.json.filter( e => ... );
```

Q4. How to populate "values" array?

Use a for-loop and find the daily confirmed cases at the corresponding date in "provinceData" array, using Array.**find**() function. If not found, put 0 into the "values" array. The number of iterations should be the length of "dates" array (The size of "values" array should be same as "dates" array.)

```
values = [];                    // create/init an empty array
let count = dates.length;       // set iteration count
for(let i = 0; i < count; i++)
{
    let date = dates[i];        // get a searching date
    // search the data from provinceData for a searching date
    let data = provinceData.find(e => ... );
    // if data exist (found), put the daily case into values array
    // if not found, put 0 instead
    if(data)
        values[i] = data.numtoday;
    else
        values[i] = 0;
}
```