

# WEB222 FINAL ASSESSMENT SUMMER 2023

---

## Overview

The WEB222 final assessment is equivalent to a final exam but can be done remotely and submitted online.

The assessment has three parts and builds on your knowledge from the course to create websites. Specifically, you will be asked to do the following:

1. Create a Multi-Page Website about hidden gems in your local area/city (**worth 12%**)
2. Implement a Static Hosting solution, so it can be viewed online (**worth 3%**)
3. Create a Reflection Video Walk-Through about the work you did in steps 1 & 2 and upload it to YouTube as an unlisted video (**worth 5%**).

## Submission

All 3 parts of your final assessment are due by **Tuesday August 15 @ 11:59 PM**. In your submission to Blackboard, please include:

- A .zip file of your complete web site code (Part 1)
- A link to your publicly hosted web site (Part 2)
- A link to your unlisted YouTube video (Part 3). NOTE: make sure you publish the video **unlisted**, so only the people who have the link can view the video

You **may** consult your notes, class recordings, and use the web, but you may **NOT** discuss any part of this final assessment with other students and may **NOT** use AI tools such as ChatGPT/Copilot to write your code: **all HTML, CSS, and JavaScript must be your own**. You may **NOT** collaborate or copy/reuse material from other sources or students.

Honesty Statement:

*I declare that my assessment is wholly my own work in accordance with Seneca Academic Policy. No part of this assessment has been copied manually or electronically from any other source (including web sites) except for the information supplied by the WEB222 instructors and / or made available in this assessment for my*

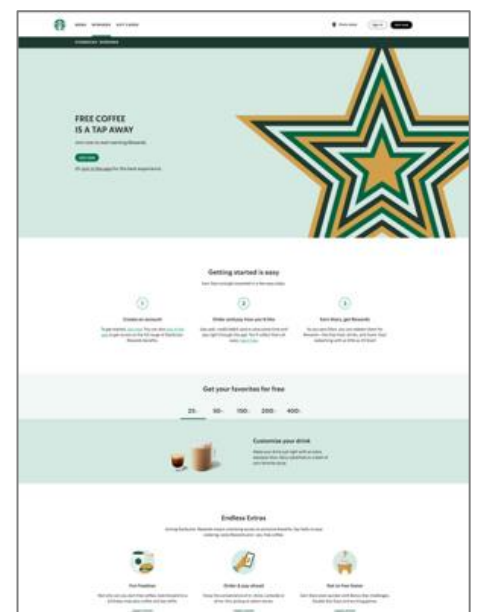
use. I also declare that no part of this assignment has been distributed to other students.

## Part 1: Website Design (12%)

You are asked to create a personal website to show off your skills as a web developer to possible employers. You are responsible for researching and implementing all aspects of this website on your own and may not include code from other students.

Your website will be expected to have the following structure and elements:

- Make sure you choose the most appropriate, **semantic HTML5 elements** for all of your page's content (e.g., don't use `<div>` or `<p>` for everything) and use them correctly. Your page must be valid HTML5.
- All CSS and JavaScript should be put in separate .css or .js files (i.e., not embedded in the HTML), and use proper indentation, formatting, and include appropriate comments.
- Your site should use elements of **Responsive Design**. That is, it should work equally well, and optimize space and sizes, on both desktop screens (more than 400 px wide) and mobile phones (less than 400 px wide). Research and use **CSS Media Queries** to define classes and rules that work on a narrow screen (400 pixels or less) vs. your full desktop (greater than 400 pixels). For example, font sizes, margins, layout choices (e.g., removing or moving elements), image sizes, etc. could all be different if viewed on a phone vs. desktop. There are lots of ways to implement a Responsive web site. Research and use a few of these techniques in order to accomplish this requirement.
- The website's pages should be divided into sections using different background colours. On the right is an example image of a website that uses this “coloured sections” design approach. Use a palette from a site like [palettelist.com](https://palettelist.com) to help pick your main and accent colours. Make sure all text uses good contrast (text colour vs. background color) for accessibility.
- Every page should use the same colours, fonts, layout, etc. At the top of each page, duplicate the same header, including navigation to all other pages. NOTE: the exact styling you use is up to you, make it



your own, but convey the same general information with the general structure:

- The site's title
  - A short description of the site
  - an image/logo. Make sure you have the rights to use any images you include
  - links to all other pages on the site
- Use **two fonts** from [Google Fonts](#) to improve the website's aesthetics: 1) for headings/titles; 2) for all other text.

**In addition to above, your website must include, at a minimum, the following features and functionalities:**

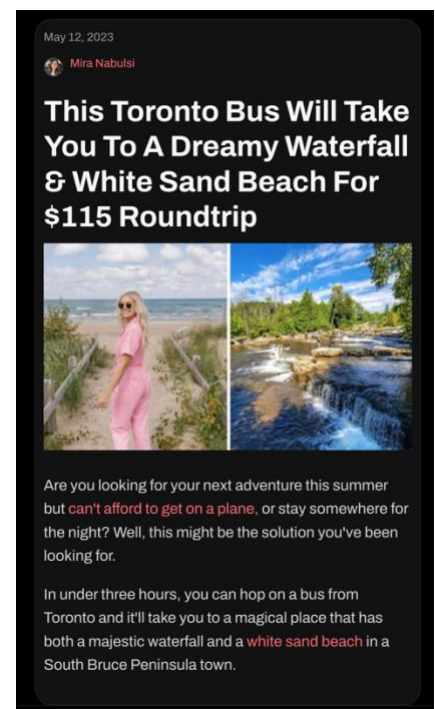
**index.html:** your main landing page. This page should give a title, short description, include some images, and provide details about each hidden gem displayed in a way that compliments your website design.

Examples:

1. <https://www.narcity.com/tag/day-trips-near-toronto>
2. <https://www.narcity.com/tag/toronto-restaurants>
3. <https://www.narcity.com/tag/things-to-do-in-toronto>

At the bottom of the page, include an **HTML Form** for the user to enter their email address and join the site's Newsletter. The form should include three inputs:

1. A single **Input** element to enter their email. Choose the most appropriate control type and attributes.
2. A hidden **Input** element with your name. It won't be shown to the user but will be submitted with the form data.
3. A "Join" **Button** that submits the form. When the form is submitted, it should POST to <https://httpbin.org/post>, and only allow the user to submit when a valid email address has been entered.



*NOTE: we are only simulating the newsletter sign-up in this assignment by creating the front-end code to accomplish it. You are not responsible for the back-end.*

**info.html:** a page that gives more detailed information about the hidden gems. It should include three sections with longer text and/or media:

1. the history of the hidden gem.
2. directions by walking, driving, and transit on how to get to the hidden gem.

3. your best suggestions on what to do/eat/see while there.

**reviews.html**: a dynamic page with other people's reviews/comments of the hidden gem. The reviews/comments should be shown as "cards" (i.e., rectangles with appropriate border, colours, text sizes, etc.), which you will generate dynamically when the page loads using JavaScript.

1. Create a file **data.js** to hold the review data. This data.js file will be an **Array of Objects**:

```
// Attach reviewData to the global window Object
// so other scripts can access it at runtime.
window.reviewData = [
  a. /* Your Review Data Goes Here... */
];
```

2. Each Object in reviewData represents a single review and should include:
  1. the name of the person
  2. the date
  3. a rating 1-5 (e.g., 3)
  4. the text of their review/comment.
3. Create at **least 5** review Objects in your data.js file. You can write them yourself or ask your friends and family.
4. When the page loads, use the review data in data.js to generate one card per review. Display the data in your review cards using the most appropriate and creative way possible. For example, the date of the review is a date, so use the most appropriate HTML element type and format. Also, the rating is a number, but you could show it visually (e.g., 3 could be shown as ★ ★ ★ ☆ ☆).
5. Below the final review, include an **HTML Form** that allows a new review to be added. Include appropriate **HTML Input elements** to enter all the necessary fields for a review and use the correct **Input Attributes**. When the user clicks a **"Create" button** at the bottom of the form, use JavaScript to add a new Object to the reviewData Array, then re-generate all the review cards (i.e., clear the cards and reshow them, including the new review just added).

*NOTE: the review added by a user won't be saved (i.e., will disappear on reload or navigation), which is OK. We are only interested in you showing it in the front-end.*

**about.html:** a page about the site and its creator. Give a short bio about yourself, why you chose these hidden gems. Include citations for any text, media, or other resources you used in the creation of the site. Finally, include the Honest Statement from the top of this assignment with your name and date.

Honesty Statement:

*I declare that my assessment is wholly my own work in accordance with Seneca Academic Policy. No part of this assessment has been copied manually or electronically from any other source (including web sites) except for the information supplied by the WEB222 instructors and / or made available in this assessment for my use. I also declare that no part of this assignment has been distributed to other students.*

## Part 2: Static Hosting (3%)

You are asked to research and implement a static hosting solution for your web site, so that it is accessible via a public URL. You do not need to spend any money to achieve this, since many free hosting services exist:

1. Netlify - <https://www.netlify.com/>
2. Vercel - <https://vercel.com/>
3. GitHub Pages - <https://pages.github.com/>

If you would like to consider registering a Custom Domain Name, that is also something that you could do. See more info at <https://www.cira.ca/ca-domains/register-your-ca>. This is optional, and not required, but good experience for all web developers.

Please submit the public URL for your project. All pages, images, etc. must work and not return 404s or other errors.

## Part 3: Reflection Video (5%)

You are asked to create a Reflection video that is 5 minutes in length and upload it as a private video to YouTube (i.e., only the people you share the URL with will be able to see it). **Make sure you test your YouTube link in an Incognito Window (i.e., not logged into YouTube as you) to make sure it will work for your professor.**

Your video should include audio of you talking and a live demo of all the elements you discuss (i.e., you don't have to be on camera if you aren't comfortable doing so, it's up to you). Use the video to give a walk-through of your website and the technologies you used.

In your reflection, please discuss and demo all of the following:

- Give a tour of your website, discussing all areas.
- Discuss the techniques, technologies, and patterns you used.
- Discuss the colours and fonts you chose.
- Which other web sites did you use as inspiration and why?
- Demonstrate and explain how you made your web site responsive, so that it works well on both phone and desktop browsers. Show how your website adjusts to window sizes less than, and more than, 400 px.
- Demonstrate the Newsletter Signup feature.
- Demonstrate how the dynamic Reviews page works, showing how to add a new review and have it appear.
- Explain some of the main features of HTML Forms and Input elements you used to help make sure the user's input was valid for your forms, and to keep them from submitting mistakes.
- How did you test that your responsive design and web form were both working, and did this testing find any bugs in your implementation? How did you address these?
- What advice would you give to other web developers building forms for their pages based on your own experience? What lessons did you learn?
- What improvements would you make to your webpage? What would you do differently if you built it again?