

User Interfaces – Lab # 2

420-2W6-AB

Lab #2 (V2 – Feb 5th)

Worth:	11%
Due:	February 14, 2023 (by 'end of day')
Hand In:	A .zip file containing your lab2 folder contents onto LEA
Late Penalty:	Late submissions lose 10% per day to a maximum of 7 days.

"Your submitted work must be clear, complete, and YOUR OWN. You must be prepared to explain any of your work to me in person. Failure to be able to defend your work, or do a similar question in front of me in person can/will void any grade you get on this lab."

Goal: To markup coherent webpage structure using HTML, and to customize website style with template and custom CSS stylesheets.

Last lab, we learned how to use various HTML content elements to create a basic website – however, we did not have very much control over the layout nor the style of our website.

In this lab, we will once again make a website about your favourite animals – but this time, we will use the correct HTML structural elements designed for defining website layouts.

Once this is done, we will see that our page layout and style can be altered significantly and flexibly with CSS stylesheet rule sets. We will use a provided CSS stylesheet for an initial layout, then make a few customizations in a separate sheet to practice CSS skills.

A full description of the required HTML/CSS changes you need to make from the provided files can be found in the [Requirements](#) section below.

Getting Started

For all web development, it is best to ensure you have an isolated working environment in your file system to organize your HTML, CSS, and other web assets. To get started, follow the steps below:

1. Create a new project folder dedicated for this assignment (lab2) in VSCode.
2. Download the provided assets .zip file ([see LEA and/or the course website](#)) and extract it inside your new project folder. You should have a folder structure like the following figure:

User Interfaces – Lab # 2

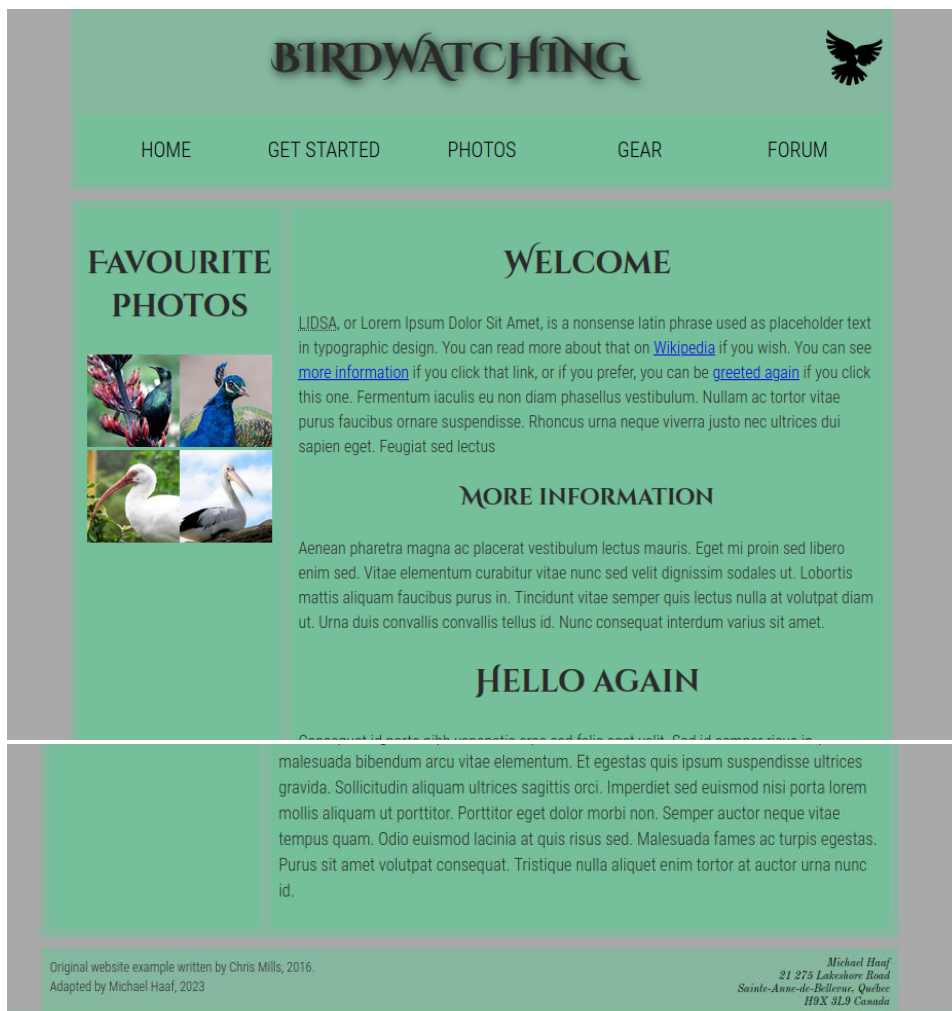
420-2W6-AB

3. You have been provided a starter **index.html** file, though it is incomplete. Your index.html will need to meet the [Requirements](#) listed below.



Once your **index.html** file is complete and correctly marked-up, by applying the the **./css/style.css** style rules it will look something like the [Example](#) below. Afterwards, you will get some practice making custom CSS rules by editing the provided **./css/custom.css** file.

Example



User Interfaces – Lab # 2

420-2W6-AB

Requirements

HTML STRUCTURE (15 marks)

For each of the requirements below, you are expected to choose the correct HTML element.

The following structure is required for your website:

1. A <link> to the default external CSS style provided
2. A header, containing:
 - a. A distinct text heading next to an icon image
 - b. A navigation menu that is a unordered list of 5 links (these links should just return the user to the same page)
3. Separated main content area, containing two columns:
 - a. A section containing at least 4 paragraphs of content and at least one <h2> and one <h3>. These headers should have **id** attributes as well as content.
 - b. A sidebar containing at least 4 image thumbnails (a small image wrapped in an anchor linking to a large image. See [w3schools.com: "How TO - Thumbnail"](https://www.w3schools.com/howto/howto_thumbnail.asp).) These thumbnail images should have appropriate **alt** attributes.
4. A footer, containing:
 - a. A paragraph containing your name/student number (see example)
 - b. An address (see example)

Note: for these requirements, **the correct layout must be achieved without making any CSS/style changes! When the content is marked up correctly, your website will automatically start to look like the example provided.**

To double clarify: **no changes should be made to the style.css file provided to you to complete these requirements.** That said, you are encouraged to study style.css to try to figure out how the styles work

Spend some time considering how the different elements of the page layout should relate to each-other: which elements go beside/inside of which other elements? It can help to draw the layout on a sheet of paper to puzzle out which elements should go where. **You can also use the Browser Dev Tools to inspect the page as you make changes.**

Once you have the layout marked up correctly, the **content** of your website should contain at least the following elements:

1. The same boilerplate as previous labs (appropriate title metadata, appropriate charset metadata.)

User Interfaces – Lab # 2

420-2W6-AB

2. A hyperlink to an external website (say, the Wikipedia article for the animal from your website). This hyperlink should have a sensible **title** attribute (so when you hover your mouse overtop, a custom message will be displayed)
3. Hyperlinks to each section header you create (when clicked, these should jump the user to the location of the section header)
4. An **abbreviation** markup so that the meaning of the abbreviation is shown when the user hovers their mouse

Beyond these requirements, you are free to make your website as thoughtful or nonsensical in content as you like (you can see in my example, for instance, that most of the website content is comprised of **placeholder** text and does not actually mean very much).

CSS STYLE (15 marks)

Once you've finished the above HTML requirements, the default CSS style should give you the same layout and style as the website seen in the Example above.

You may have noticed that there were two CSS files provided: one of them is there for you to create a custom style.

In order to start using the new custom.css file, insert the following HTML snippets into your **index.html** file:

<https://codepen.io/michaelhaaf/pen/abjjKew>

NOTE: The button will not be visible at first when you paste it into your website. I have put updated instructions into the HTML snippet to troubleshoot this problem.

You can now preview the effect of the alternate style by clicking on the provided button in your browser. At first, it will do nothing. Try adding the following code snippet to the **css/custom.css** file

Now it's time to make a custom style. You should apply at least the following changes:

1. Make all headings use **an external, serif webfont** with ``Georgia`` and ``serif`` as fallbacks
2. Make all paragraphs use **an external, sans-serif webfont** with ``Verdana`` and ``sans-serif`` as fallbacks
 - a. For requirements 1 & 2, <https://fonts.google.com/> is a good source of external webfonts.
3. Make the level one headings pink, using the CSS color keyword hotpink.

User Interfaces – Lab # 2

420-2W6-AB

4. Give level one headings a 10px dotted border-bottom with a color specified in the RGB format.
5. Make level two headings italic.
6. Give all odd-numbered p elements a blue background and white text using HEX color codes.
7. Give all `p` elements that are descendants of the `footer` element a font-size of 10px
8. Using pseudo-classes, make all links red (using HSL colors) and bold when hovered
9. Using pseudo-classes, make all nav bar links have a red background when clicked.
10. Using a “class” attribute (you will need to edit the index.html file), apply text-shadow to all text elements you give a certain class.

You may make whatever other changes you wish, as long as the above 10 requirements are met.