CS422 User Interface Design and Programming
University of Illinois Chicago
Instructor: Nikita Soni, Ph.D.

HTML + CSS Lab Assignment

Due: January 25th, 6:00 pm
Total points: 20

Document structure

- **Introductory Materials**
  - TA Video Session + Slides + Software
  - What is HTML+CSS?
  - Lab Assignment Setup + Additional Resources
- **Lab Assignment: Responsive Online Portfolio (20 points)**
  - Assignment Details
  - Deliverables
  - Rubric

Introductory Materials

TA Video Session + Slides + Software

The presentation slides from the TA video session can be found here; TA video session is here. The only software you'll need for this lab is **Google Chrome** and a **text editor**, such as Sublime, Atom, or VSCode.

What are HTML + CSS?

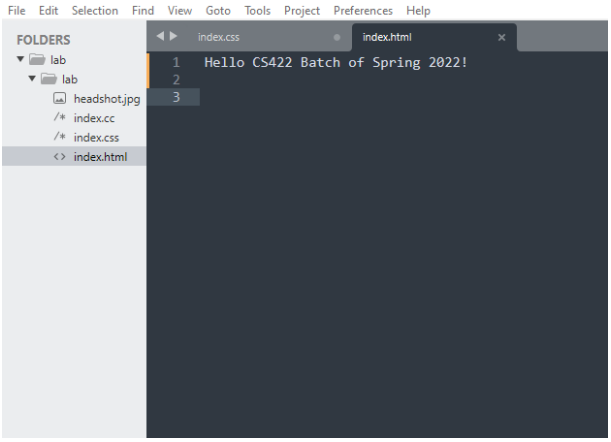
HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are coding languages that web browsers use to understand how to display a page and what it should look like in terms of content and style. HTML is used to lay the foundation and overall structure, and CSS is used for styling, like colors and fonts. You can think of HTML like the blueprint of a house, and CSS like the interior decorating scheme. Web pages only need HTML, but they look pretty plain without CSS. Every website is built using HTML and CSS, so you've already had tons of interactions with them!

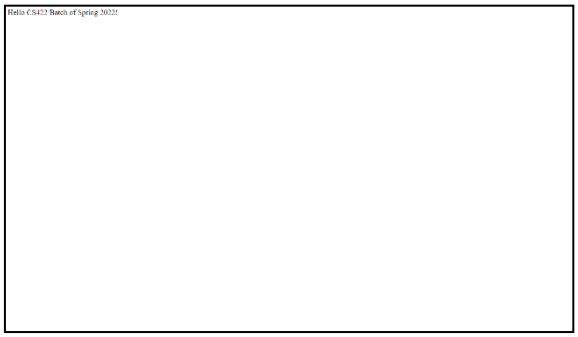
Lab Assignment Setup + Additional Resources

To start, create a folder on your desktop called **lab**. This is where you be storing all the files you make for this assignment to stay organized.

Next, let's open up a text editor. Create a new file and type Hello CS422 Batch of Spring 2022! on the first line. Save this file in the lab folder as *index.html*. Awesome, you've just

made your first HTML file! If you open the lab folder and double click *index.html*, a new page should pop up in your web browser that looks something like this:





Your computer's default way of opening .html files is to display them in your web browser. It's just like when you double click on a file ending in .doc and Word automatically opens up. This also means that if you want to edit an HTML file, you'll need to right-click to open it in a text editor.

Basic Tags

Now, let's start writing some more HTML. Reopen *index.html* in a text editor, delete "Hello CS422 Batch of Spring 2022!", and then replace it with **<!DOCTYPE html>.** This lets the browser know that everything in the file after that line will be in HTML. Most of the basic tags were already covered by the TA in the video posted on Blackboard. But let's do this step-by-step exercise together to get you started with the assignment.

HTML is made up of different **tags** that are used to label different parts of the page so that later we can style and adjust them. We always start off with an `html` tag, so on the second line type in `<html>` and on the third line type in `</html>`. Often, tags in HTML come in pairs, so the first `<html>` on line two is called an **opening tag**, while the second `</html>` on line three is called a **closing tag,** which is why we added in the forward slash.

Inside the html tags, type in the opening and closing `<head>` and `<body>` tags. Everything between the head tags is information for the browser to understand your page (the **metadata**), and everything between the body tags is content that the user sees. Inside the head tags type in an opening and closing `<title>` tag. So far, you should have something like this.

```
<!DOCTYPE html>
<html>
    <head>
        <title> My Portfolio </title>
    </head>
    <body>
        Hello, I am Nikita Soni. Welcome to my Portfolio!
    </body>
</html>
```

Save *index.html* and refresh your browser. This will give your page a name that your browser will display in the tab.

While content can be placed directly within the body tags, there are other tags that can be used to structure content on the page. For example, header tags (`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`) can be used to create headings on a webpage. The tag `<h1>` creates the largest of these headings, while the tag `<h6>` creates the smallest of these headings, with each header tag from `<h1>` getting progressively smaller.

The `<p>` tag defines a paragraph. Paragraphs are typically represented in media as blocks of text separated from other blocks by indentation, but HTML paragraphs can be any structural grouping of related content, like images or tables.

Next, write your name inside an `<h1>` tag in index.html. Then, create a `<p>` tag and fill it with the lorem ipsum text below. Make sure to put this within the `<body>` tags!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

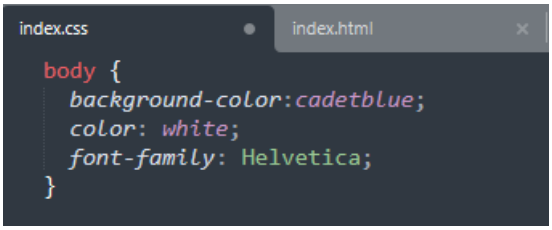
Lorem ipsum is a type of placeholder text often used in graphic design.

Again, save *index.html* and refresh your browser.

Linking Files

The webpage you have right now is okay, but it could definitely use some color. So, we'll start writing some CSS. Create a new file in a text editor, and save it in the lab folder as index.css. In CSS, we use the names of the tags from our HTML file to indicate where we want to apply our styles. In your CSS file, type `body {}`. This means that any style properties inside the curly brackets will be applied to objects in the body tag. Inside the curly brackets,

we'll have a list of properties and that values we want them to have. We'll end each pair with a semi-color for punctuation.



Now, save everything and refresh the page. Nothing has changed, right? That's because we need to create a link to our CSS file in the head of our HTML file. Between the `<head>` tags, type out `<link rel="stylesheet" href="index.css">` so that HTML file has a link to the CSS file. Save index.html and refresh the page again. You should now have a color background and styled text!

Styling

CSS IDs are unique names that we give to tags to be more specific. This can be done by adding the id attribute inside the tag. Attributes will generally follow the format: `name="value" (e.g., id="mypic").`

To apply CSS styles to a specific element, we can reference their id in index.css. This is similar to referencing an HTML tag, but to refer to an id, you'll need to add a pound sign `(#)` in front of the name of the id in index.css before setting styles.

Note: To apply CSS styles to a set of specific elements, we can use classes. Classes are like IDs, the only difference is that they can be given to more than one element. To refer to a class of elements, you'll instead need to add a period (.) in front of the name of the class in index.css before setting styles.

Below are the definitions of several common CSS properties which you might find helpful for the assignment.

- **`margin`** gives elements space *outside* the border of elements
- **`padding`** gives elements space *inside* the border of elements
- **`width`** and **`height`** set fixed dimensions of our boxes
- **`Max-width`** and **`max-height`** set limits to the size of an object.
- **`background-color`** sets the background color of the element
- **`color`** sets the color of all text inside the element
- **`font-family`** specifies the font for an element
- **`float: left`** tells our elements to be as far to the left as possible without

```
    overlapping with one another.
```

These are only a fraction of the available properties in CSS, but provide a strong foundation for styling moving forward.

Positioning

What should we do if we want to move multiple boxes together? One way to solve this is to wrap divs of two boxes around another *div*, which serves as a container.

Using this container, we can center our smaller divs in the middle of the page together. In *index.css* we need to first make sure the container is large enough to fit the contents within it, and then make sure to center it on the page.

Images

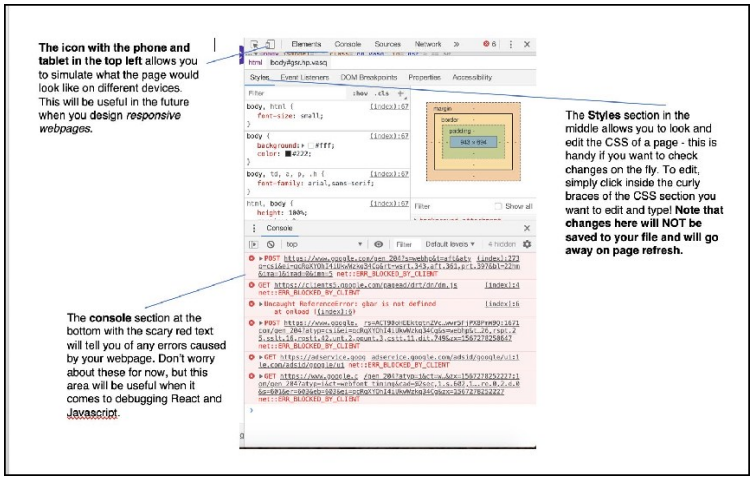
When adding images in HTML, you can either add a local file or a direct link to an image on the internet. Note: If you're working locally, make sure you put the image in the same folder as your HTML + CSS files — it'll make referencing it easier.

Linking Pages

On websites, there are typically links to other pages that help us navigate to other pages. To create a link in HTML, we use the <a> tag and wrap it around the text we want to be our link. The <a> tag has a property called href, which is where we'll put our link.

Chrome Developer Tools

While you're building your webpage, you can use Chrome's Developer Tools to take a look at the HTML and CSS of a page. Additionally, Dev Tools will also help you identify any errors in your code and even let you see what changes to your HTML and CSS code would look like! Getting familiar with Dev Tools will definitely save you a lot of time and effort in the future.



Additional Resources

If you want to learn more, we have some resources listed below.

- HTML Dog - HTML
- HTML Dog -CSS
- w3schools
- codecademy
- Dash by GeneralAssembly

CSS Flexbox + CSS Grid

When it comes to more complicated layouts, CSS offers some more tools to make organization much easier. The two techniques are **CSS Flexbox and CSS Grid**.

CSS Flexbox is a CSS library that makes it easier to lay items on a page nicely and have it be responsive. CSS Grid is another library that many developers use for establishing a responsive layout. Flexbox is usually used for smaller-scale layouts, while Grid is typically used for larger-scale layouts. One of the strengths of Grid and Flexbox is that they allow your page to be responsive — meaning that it will look nice on varying device sizes.

- ➢ Intro to CSS Grid
- ➢ Intro to Flexbox

For resources on learning how to use Grid and Flexbox, check out the links above! If you use any of these resources, cite them in the Readme file.

CS422 User Interface Design and Programming
University of Illinois Chicago
Instructor: Nikita Soni, Ph.D.

Lab Assignment: Responsive Online Portfolio (20 points)

Overview

Maintaining an online portfolio or a personal webpage that discusses interface design projects is a great tool for those considering careers in UX/UI design. In this assignment, you will design a stencil code structure of a responsive online portfolio using HTML + CSS. Of course, you can also use other free website hosting services (e.g., wix.com) to maintain your online portfolios, but designing on your own will give you more control over your webpage.

Specifications

This assignment will help you get started with building a **stencil code structure** for an online portfolio if you do not already have one. You will create the elements in HTML, and specify the style and layout with CSS. Write the HTML and CSS by hand. For responsiveness, you should only use CSS Grid and/or Flexbox. Grid is better for interfaces with elements laid out all over, while Flexbox is better for single-column interfaces. **Do not copy and paste code from elsewhere.** Your code should contain comments to enhance readability. You will lose points for bad code readability.

Description

By following the lab setup discussed above, by now you should have a basic structure of your webpage set up. For this assignment, you will create a **responsive webpage** that looks like the one below. Your submission will be evaluated for **structure, layout, and responsiveness on 3 screen sizes**: a small phone in portrait orientation, a tablet in both portrait and landscape orientations, and a widescreen desktop monitor. When necessary, your webpage should shrink or expand elements, wrap to the next line or disappear in some cases, or move relative to the screen edges.

Webpage Components

Your webpage should have the following working components:

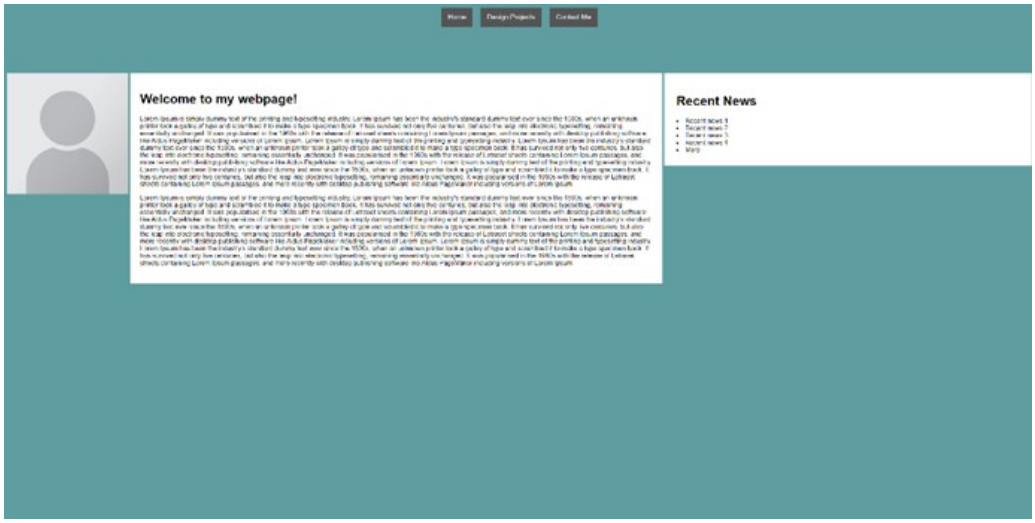
- The **title** of the page should be “My Online Portfolio”
- The page should have **3 navigation bars**: **Home**, **Design Projects**, and **Contact Me**. Your assignment includes designing **two** of these pages: Home and Design Projects page.
    - The **Home page** (see image below) should be divided into three columns containing: a *Headshot image* (sample image can be found here), *Welcome text* (2 paras, sample text) as shown in the image below, and a *Recent News list*. You can use colors and media/icons of your choice, but make sure to use the provided sample text.
    - The **Design Projects page** (see image below) should list *four project placeholders*. Each placeholder represents a design project in your portfolio and should have a *reference image* (sample image, can be found here) and *3-*

line dummy project text (sample). Note that for project placeholders; image and text holders are attached and do not get detached when screen size changes. You can use colors and media/icons of your choice, but make sure to use the provided sample text.

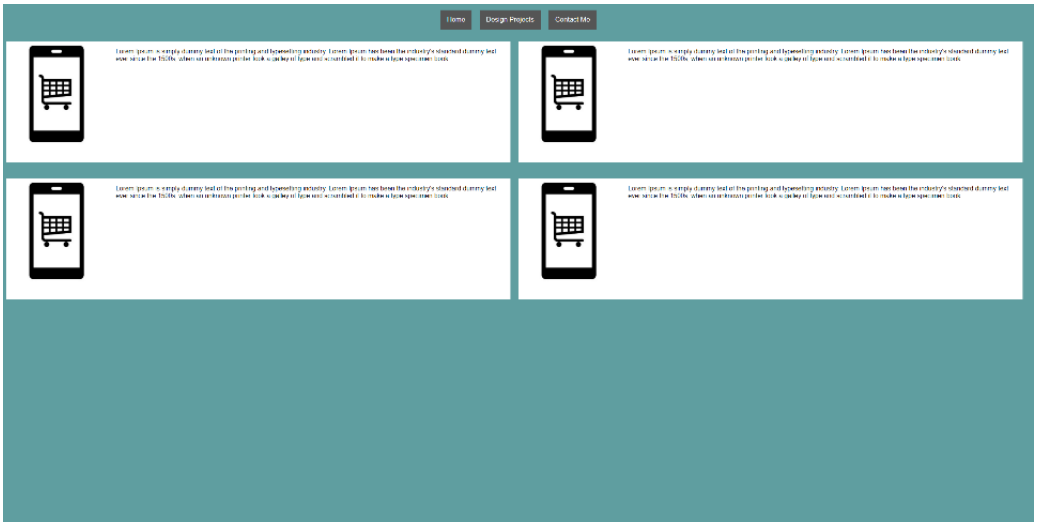
- **Demo** of how your responsive webpage should be working like: [link]

Possible approaches or tips: you can consider using a flexbox container with three divs for the Home page. Media queries (as discussed in the TA video session) can be used to adapt your design to different screen sizes and resolutions.

Home (Desktop Version)



Design Projects (Desktop Version)



Deliverables

- All submissions should be made via **Blackboard** on **January 25th, by 6:00 PM**
- The zip file of the lab folder (lab.zip) containing the following items:
    - **Index.html**
    - **Index.css**
    - **designprojects.html**
    - **images** used in the project
    - S**creen capture** of the interface showing responsive UI testing as done in the video.
    - **Readme** file containing citations to any sources you might have found helpful.

Rubric

- The layout and structure of the Home webpage is similar to the example screenshot in the assignment (4 points)
- The layout and structure of the Design Projects webpage is similar to the example screenshot in the assignment (4 points)
- The webpage is responsive and works well with 3 screen sizes: a small phone in portrait orientation, a tablet in both portrait and landscape orientations, and a widescreen desktop monitor (8 points)
- HTML and CSS files contain comments for better readability (2 points)
- Screen capture of the project showing responsive UI is present (2 points)

Course Policies

- All students should follow standards of conduct as discussed in UIC's disciplinary policies (see syllabus for more details). As per the course policies, you should be doing this assignment independently without discussing it with other students in the class or elsewhere. Do not copy and paste code from online sources.