Creative Project 1 Specification

Due: Friday, April 9th 2AM PST (**morning**)

Overview

For your first Creative Project, you will use what we're learning about HTML and CSS to make a simple website with at least two HTML pages linked to one shared CSS file. As a Creative Project, you have the freedom to have more ownership in your work, as long as you meet the requirements below.

As a reminder, assignments will alternate between creative projects (CPs) and more formal homework assessments (HW sets). We have designed assignments to support each module of this course, and Creative Projects as exploratory opportunities to prepare you for the following HW in each module and get more direct help and feedback on things like correct use of new languages and technologies.

We encourage you to explore the new material covered in class, as well as related (but optional) content we may link to along the way (e.g. CSS3 animations), as long as you follow the code quality guidelines. This is your chance to:

1. Build websites that you can link to on your resume or code portfolio (CPs can be public, most HWs cannot be).
2. Ask Melissa and/or TAs about features you want to learn how to implement (we can provide more support for CPs than HWs) and ask for feedback/ideas on Discord.
3. Apply what you're learning in CS 101 to projects you are personally interested in and may use outside of just a CS 101 assignment.
4. Get feedback on your use of new languages and technologies we're learning that will be required in the following HW set, which will be worth more points.
5. Optionally showcase your work on the course CP showcase!

You may choose to do a new website for each CP, or build on a single project, as long as you meet each CP requirements (note that you do not need to meet any CP1 requirements in later CPs).

Ideas for CP1

As long as you meet the requirements outlined below, you have freedom in what kind of website you create. Here are some ideas (the staff is more than happy to help discuss more ideas on Discord or during office hours!):

- Create an About Me page
- Turn your current resume into a webpage (make sure you still have a second HTML page to link to that resume)
- Implement a simple website for a club or organization you are in
- Write a website for a friend/family member
- Write a website with facts on your favorite animal/hobby/topic
- Write a random website about a random thing (and be creative!)
- Write a "tutorial" website on the basics of HTML/CSS given what you're learning (teaching is a great way to reinforce what you're learning!)
- Start a blog website, perhaps documenting what you're learning this quarter in one of your classes
- Showcase any work about a hobby you may have (art, 3D printing, sports, travel, catspotting, puzzles, etc.)

External Requirements

- Your website must contain at least two HTML pages that are "linked together" - in other words you can use a link to get from one page to another, and a (relative) link to get back to the first page.
- A `wireframe.png` or `wireframe.pdf` sketch (either hand-drawn or digital) of at least one .`html` page outlining major page sections (you are encouraged to include annotations for styles, etc. in this process, but it's not required). This is good practice for web developers to approach page content, structure, and layout before writing HTML/CSS from scratch, and will be a useful strategy in larger assignments. You are welcome to submit more than one (these can be great to include for a web development portfolio project), and will be graded on completion, not design or quality. Refer to Lecture 2 slides for an example.
- You must link `styles.css` to both HTML pages to style your website with a consistent look and feel. You may add a second .`css` file to either page to factor out styles that are not shared by both pages (you can add a second CSS file with an additional `<link>` tag in the HTML `<head>`). If you choose to use a second stylesheet, you should use styles.css **only for styles that are shared by both HTML pages**. This is good practice to improve website maintainability. **Do not include any inline CSS in your HTML.**

- **In at least one .html file**, you must use at least 8 **different** types of HTML tags total in the `<body>`, in addition to the required `<!DOCTYPE html>`, `<html>`, `<head>`, `<title>`, `<link>`, and `<body>`)
- Suggestion: Refer to this page for a comprehensive list of different HTML tags, and post on Discord if you have any questions about any! You may use ones we haven't talked about in lecture, since there are many more that we could possibly cover in class as long as they are not in the list of deprecated tags from this page and they are used with appropriate semantics (e.g. only use `<table>` for tabular data, not to force layout of columns of text).
- At least 2 of the 8 tags should be semantic tags listed under "HTML Layout Elements in More Detail" here.
- `styles.css` must have:
    - at least 4 additional different rulesets. Refer to this page for a CSS reference of selectors. One of your rulesets must use a combinatorial selector, and one of your rulesets should have a grouped (comma-separated) selector.
    - at least 5 different CSS properties defined which style content in your HTML files.
    - at least one Google font of your choice imported and used with an appropriate default font (e.g. sans-serif) specified. Remember to import Google fonts at the top of your CSS file, not in the HTML!
- **All file names and links in your project must be lower-cased without spaces** (e.g. `img/puppy.jpg` but not `img/puppy.JPG` or `img/Puppy.jpg`). This is enforced to avoid broken links commonly occurring in CP/HW submissions due to case-insensitivity of file names on Windows machines. In general, it is also just good practice for file/directory naming.
- Your page should include appropriate content and copyrights and citations. If we find plagiarism in CPs or inappropriate content, **you will be ineligible for any points on the CP**. Ask if you're unsure if your work is appropriate.

Internal Requirements

- Links to **your** .html and .css files should be **relative links**, not absolute. It is fine to use absolute links to other websites/images.
- Your HTML and CSS should demonstrate good code quality as demonstrated in class and detailed in the CS 101 Code Quality Guidelines. You are expected to follow all the requirements listed under HTML and CSS (ask if you have any questions!). Common code quality requirements students miss include:
    - Using consistent indentation, proper naming conventions, curly brace locations, etc. Remember that IDs and classes should be in all-lowercase conventions and multiple words are optionally separated by "-". CSS rule sets should be separated by a single blank line and use indentation appropriately.

  - Keep lines fewer than 100 characters for readability (links are an exception to this rule)
  - Do not express style information in the HTML, such as through inline styles or presentational HTML tags such as b or font.
  - Prefer CSS selectors instead of using too many classes or IDs in your HTML.
  - Do not include unused, duplicate, or overridden CSS rules or rulesets and use shared CSS selectors to factor out redundancy (see Code Quality Guide for more examples). Make sure to double-check that you didn't leave any unused styles in before submitting!
- Note: You *may* use a framework such as Bootstrap to help with your styling and helpful responsive layout features, however you must still meet all of the above requirements and demonstrate that you understand the key concepts of how the HTML and CSS work. Any framework code *will not count* towards HTML/CSS requirements (e.g. if you use the Bootstrap "container" class in your HTML, you cannot count the CSS implementation the `bootstrap.css` file towards the CSS requirements), however you can add new (not duplicate) CSS for this class to `styles.css`. You are not allowed to use any template HTML files for frameworks (this defeats the purpose of writing HTML and CSS from scratch in this first assignment).
  - Don't know what any of that means but want to learn how to use a CSS framework? Ask about them on Discord or during Office Hours!
- HTML and CSS files must be well-formed and pass W3C validation (this will be graded). To validate your HTML, copy/paste your HTML code into the textarea option on the page. Warnings are generally ok, as long as they are not violating a requirement Melissa has clearly specified.

Documentation

Place a comment header in each file with your name, section, a brief description of the assignment, and the files contents. Examples:

HTML File:

```
 <!--
     Name: Lorem Hovik
     CS 101 Spring 2021
     Date: April 1st, 2021
     This is the index.html page for my portfolio of web development work.
     It includes links to side projects I have done during CS 101,
     including an AboutMe page, a blog template, and a cryptogram generator.
-->
```

CSS File:

```
 /*
    Name: Lorem Hovik
    CS 101 Spring 2021
    Date: April 1st, 2021
    This is the styles.css page for my portfolio of web development work.
    It is used by all pages in my portfolio to give the site a consistent
    look and feel.
*/
```

Grading

Our goal is to give you feedback, particularly on the internal requirements and style and documentation, so you can incorporate this feedback in your homework assignments which will be worth more towards your final grade.

This CP will be out of 9 points and will be distributed as:

- External Correctness (4 pts)
- Internal Correctness (3 pt)
- Documentation (1 pt)
- Wireframe (wireframe.png or wireframe.pdf) (1 pt)