**Unit 01 HTML, CSS, and Git Homework: Portfolio**

Build a professional portfolio site using HTML and CSS following a set of specifications and a style guide.

**User Story**

AS A web developer

I WANT a deployed portfolio

SO THAT I can showcase my applications to potential employers and clients

**Business Context**

A portfolio is often the first point of contact for employers looking to fill a position or clients in need of a developer to complete a project. A clean and informative portfolio showcases a developer's strongest work and can lead to employment and/or more clients.

**Acceptance Criteria**

GIVEN that an employer or client wants to hire a developer

WHEN viewing their portfolio

THEN they are able to learn about the developer, find contact information, and follow links to projects and repositories

**Instructions**

* Create a new GitHub repository, initialize it with a README.md , and clone it to your local development environment.
  + ❗ Follow best practices for your README as outlined in the Good-README-Guide
* Inside the repo folder on your computer, create the following:
  + 3 HTML documents: index.html, contact.html, and portfolio.html.
  + A folder called assets.
  + Inside the assets directory, make two additional folders: css and images.
    - In the css folder, make a file called style.css.
    - In the images folder, save the images you plan on using (like your profile image and the placeholder images for the portfolio).
* Refer to these screenshots for your site:
* **IMPORTANT** Your site's layout must match the designs in these screenshots. You'll find the specs for these designs below.
* When you create a website, it's standard practice to code your HTML files before writing any CSS and to complete one HTML file before moving onto the next. Write your HTML semantically too:
  + If something is a heading, use a heading element.
  + If something is a list, use an unordered or ordered list element.
* The content in index.html should be unique to you in the following ways:
  + Write a paragraph or two about yourself. Make it fun; show your personality!
  + The main logo where it says "Your Name" should say your name. This may make the logo section smaller or wider depending on how long your name is—that is fine and expected.
* After all of your HTML is written, you can begin styling your pages using the style.css file you created.
  + Be sure to [validate your html](https://validator.w3.org/#validate_by_input).
  + Consult the following Style Guide section for advice on styling your CSS files.

**Style Guide**

* Colors (**Pro Tip:** Use the [Eye Dropper](https://chrome.google.com/webstore/detail/eye-dropper/hmdcmlfkchdmnmnmheododdhjedfccka) Chrome extension to find the colors that are used on webpages.)
  + Teal color (used for headings and backgrounds): #4aaaa5
  + Regular font color (used for paragraphs and all text besides the headings): #777777
  + Main header background color: #ffffff
  + Main header border color: #cccccc
  + Footer background color: #666666
  + Main content background color: #ffffff
  + Main content border color: #dddddd
* Fonts:
  + For heading fonts use font-family: 'Georgia', Times, 'Times New Roman', serif;
  + For all other fonts use 'Arial', 'Helvetica Neue', Helvetica, sans-serif; s
* Profile image in index.html:
  + Use a picture of yourself.
* Portfolio images:
  + Save the images to your images folder.
* Background images:
  + The background pattern used was found on [Subtle Patterns](https://subtlepatterns.com/). You can browse through that site and use whichever pattern you like.
* Dimensions:
  + The entire content and the main section content area is 960px wide.
* Make a "sticky footer." You will notice that the dark gray footer will always rest just below the content. This is fine whenever your site has enough content to push it down past the height of most monitors. But if there isn't much content in the body, the footer could rest in the middle of the page:
  + Try to code the footer in a way that it will always remain at the bottom of the page, no matter how short the content is. Refer to this [sticky footer tutorial](https://css-tricks.com/couple-takes-sticky-footer/) for guidance.
  + Give the footer the following CSS: border-top: 8px solid #4aaaa5;
* Look into these concepts: float, padding, margin, display, overflow, clear and text-align.

**Commit Early and Often**

One of the most important skills to master as a web developer is version control. Building the habit of committing via Git is important for two reasons:

* Your commit history is a signal to employers that you are actively working on projects and learning new skills.
* Your commit history allows you to revert your codebase in the event that you need to return to a previous state.

Follow these guidelines for committing:

* Make single-purpose commits for related changes to ensure a clean, manageable history. If you are fixing two issues, make two commits.
* Write descriptive, meaningful commit messages so that you and anyone else looking at your repository can easily understand its history.
* Don't commit half-done work, for the sake of your collaborators (and your future self!).
* Test your application before you commit to ensure functionality at every step in the development process.

We would like you to have well over 200 commits by graduation, so commit early and often!

**Submission on BCS**

You are required to submit the following:

* The URL of the GitHub repository