U DACITY

PROJECT SPECIFICATION

Personal Blog Website

Structure

CRITERIA	MEETS SPECIFICATIONS
CSS Separate From HTML	1. Portfolio completely separates structure from design/style. 2. There are no attributes present in the body of the document. 3. There are no elements in the document.
CSS Imports	There should be at least 3 imported files in the main CSS file, but the student is welcome to break it down even further if that makes it easier for them.
Directories & Files	Files are organized with a directory structure that separates files based on page and functionality.
Links	There is an intentional user flow on each page with appropriate links as needed.

Design

CRITERIA	MEETS SPECIFICATIONS
CRITERIA	MEETS SPECIFICATIONS
Custom Design	Custom images, layout, and styling.

Foundations/Building Blocks	• **Typography**: Custom design for typography with at least 3 unique properties for each typography selector
	
	  1. Headers (h1 to h3 at minimum) `<h1>`
	  2. Paragraph Text `<p>`
	     a. Bold
	     b. Italic
	     c. Underlined
	  3. Links `<a>`
	  4. Quotes
	
	• **Colors**: At least 3 colors are used.

Components	The following are used in the webpage: • Image(s) • Image caption • Buttons • Card

CRITERIA	MEETS SPECIFICATIONS
Patterns	The general pattern of the webpage should be as follows:

- **Blog Homepage**

  - Navbar
  - Blog Post Card

    - Card
    - Image
    - Buttons

  - Footer

- **Blog Post**

  - Header
  - Author Info

    - Name
    - Title
    - Current Company/School
    - Short Bio
    - Avatar

  - Social Sharing

    - Twitter
    - Facebook
    - LinkedIn

CRITERIA	MEETS SPECIFICATIONS
Pages	The following pages should be present: 1. Blog Homepage 2. Blog Post

Layout

CRITERIA	MEETS SPECIFICATIONS
Flexbox Based Layout	Uses Flexbox at least twice throughout the project to create a component or pattern
Grid Based Layout	Uses CSS Grid at least twice throughout the project to create the layout for pages

Responsiveness

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Multi-Device Web Design	Pages are mobile-friendly and display correctly on all display sizes (mobile, tablet, desktop). Note: You can simulate Mobile Devices with Device Mode in Chrome DevTools.

Quality

CRITERIA	MEETS SPECIFICATIONS
Valid HTML	- HTML5 semantic tags such as `<header>`, `<footer>`, `<article>`, `<section>`, etc. are used to add meaning to the code. - No div or section selectors are without a CSS class or id.
HTML Formatting Rules	- All code is lowercase - The code does not have trailing white spaces. - Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc). - Code uses a new line for every block, list or table element and indent every such child element (it is acceptable to put all elements in one line). - When quoting attribute values, code uses consistent quotation marks (single vs. double).

CRITERIA	MEETS SPECIFICATIONS
HTML Style Rules	- HTML documents HTML5 `<!doctype html>` - Code omits type attributes for style sheets and scripts. [Optional] Code does not use entity references unless necessary e.g. characters with special meaning (like `<` and `&`) as well as control or "invisible" characters (like no-break spaces).
CSS Formatting Rules	- The code does not have trailing white spaces. - Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc). - Code indents all content, that is rules within rules as well as declarations to reflect hierarchy and improve understanding. - The code uses a semicolon after every declaration for consistency and extensibility reasons. - Code always uses a space after a property name's colon, but no space between property and colon, for consistency reasons. - Code always uses a single space between the last selector and the opening brace that begins the declaration block. Code always start a new line for each selector and declaration. - Code always put a blank line (two line breaks) between rules. - Code uses consistent quotation marks for attribute selectors or property values (single vs. double).

CRITERIA	MEETS SPECIFICATIONS
CSS Style Rules	• The code uses meaningful or generic ID and class names that are as short as possible, but as long as necessary. • The code does not use element names in conjunction with IDs or classes. • The code uses shorthand properties where possible. • [Optional] Code omits unit specification after 0 values. • [Optional] Code includes leading 0s in decimal values for readability. • [Optional] Code uses 3-character hexadecimal notation where possible. • [Optional] Code separates words in ID and class names by a hyphen.

Suggestions to Make Your Project Stand Out!

- **Custom Text:** You are welcome to use Lorem ipsum or other placeholder text in your blog and blog posts, but putting in real, custom text can make your project look even more impressive!
- **Pattern - Related Articles:** Add 2 to 3 fake blog post references after the blog post ends (i.e. to older blog posts)
- **Pattern - Subscribe Form:** Instead of a fake subscribe link on the blog, there is an actual form area with name and email input fields and a subscribe button.
- **Project URL:** Project is fully deployed on GitHub Pages, Netlify or another hosting provider that is free of charge.
- **Design System:** Consider implementing a design system for your webpage.
- **Accessibility:**
    - Discernable text
    - Sufficient color contrast
    - ARIA attributes

- ARIA attributes
  - Less than 5 violations found through the Axe (Chrome extension) test on the
    web page.