Rubric

Use this project rubric to understand and assess the project criteria.

Basic Requirements

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Follow wireframes to build a project layout | * The page structure matches the provided wireframes. * The logo (if included) is at the opposite end from the navbar. * Proportions, alignment, and placement of elements are consistent with the design. |
| Create a responsive design that adapts to different screen sizes | * Pages adjust appropriately for desktop, tablet, and mobile views. * No elements overflow or break the layout when resizing the browser window. |

Methodology

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Organize files with a clear structure based on BEM methodology | * The scss/less folder structure includes separate folders for base, blocks, and utils. * Each folder has a main stylesheet file matching the folder name (e.g., base.scss, blocks.less) unless using the **granular approach**, in which case, no utils file is needed. * Block names reflect the component's purpose (e.g., header, button, bio) rather than being descriptive (e.g., large-text). * If using the **aggregated approach**, a utils.scss or utils.less file consolidates utilities like \_variables and \_mixins for import into blocks. * If using the **granular approach**, \_variables and \_mixins are imported directly into block files as needed. |
| Apply the BEM methodology to structure CSS | * Class names that are created in preprocessor files use BEM methodology * Class names are logical and consistent, reflecting the page's structure   You're welcome to refer to the official BEM Methodology documentation to review the BEM rules: [**https://en.bem.info/methodology/quick-start/(opens in a new tab)**](https://en.bem.info/methodology/quick-start/) |

Preprocessors

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Compile preprocessor stylesheets into a primary CSS file | * The /dist folder contains a main.css file which is compiled using a preprocessor through console commands. |
| Use variables and mixins for reusable styles like colors, fonts, and spacing | * Variables are defined for reusable values (e.g., $primary-color, @font-size-base). * Variables are applied consistently across SCSS/Less files. * At least one mixin is created for common patterns (e.g., media queries, reusable styles). * Mixins are used effectively across SCSS/Less files. |
| Apply nesting to organize related styles hierarchically | * The BEM element and modifier selectors are declared in a nested format to avoid repetition:   + A .block selector should contain all the related &\_\_element selectors.   + Modifiers should be nested under their respective blocks and elements, using the &\_modifier syntax (e.g., .block\_\_element and .block\_\_element\_modifier).   + Avoid deep nesting to maintain readability and ensure maintainable code. |

CSS Techniques

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Implement advanced CSS properties | At least one of the following advanced CSS properties is present in the project:   * Math Functions (calc) * Scroll Snap * Inset text color * Anchor * Color Scheme Media Query * Backdrop Filter * Min and Max |
| Enhance interactivity with CSS animations and transitions | * The buttons change background color when hovered over or when clicked/tapped, and this color transition is smooth. * Includes at least one other transition or animation technique, such as:   + Hover transitions (in addition to the buttons, something that affects properties other than color)   + Animations defined with @keyframes * Effects are smooth, non-distracting, and align with the design intent |
| Incorporate responsive animations and dynamic effects | * The nav header needs to minimize its height as the user scrolls down. * In addition to the nav, the project needs to implement at least one more of the following responsive or dynamic animation techniques:   + On-scroll effects   + Animated backgrounds   + Turning off animations via media queries |

Accessibility

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Ensure proper semantic HTML and relationships | * Webpage uses semantic tags correctly (header, footer, main, section, h1-h6). * Heading tags descend in the correct order. * Labels are associated with input fields using the for attribute (if applicable). |
| Provide accessible non-text content | * All images include meaningful alt attributes (or alt="" for decorative images). * Icons or links without visible text use aria-label or are wrapped in a visually hidden class (e.g., sr-only). * Form buttons have descriptive text or aria-label attributes. |
| Ensure keyboard and focus accessibility | * All functionality is accessible via the keyboard (e.g., tabbing through links and buttons). * Focus order is logical and intuitive. * Keyboard focus never gets stuck (no keyboard traps). * Focused elements are visibly highlighted. |
| Use color and sensory characteristics accessibly | * Color is not the sole method to convey information. * Links are distinguishable from surrounding text without relying solely on color. * The contrast ratio between text and background is at least 4.5:1. |

Suggestions to Make Your Project Stand Out

**Intro animations:**  You can have elements animate into existence by fading in or sliding into place once the page loads.  
  
**Other sections:** The suggested navbar items are pages and/or components that one might find in a typical portfolio website. Although a thorough list of projects and a resume should have their own pages, sections such as Skills and Contact could be included on the home page. Consider building them out into their own sections.  
  
**Custom background images**: Custom background images can be found at [**https://www.pexels.com(opens in a new tab)**](https://www.pexels.com/). They are free and don’t require purchasing or a license. They can also help inspire your overall color theme.