**Challenge Project: Build a Website Design System**

**Overview**

This project is slightly different than others you have encountered thus far on Codecademy. Instead of a step-by-step tutorial, this project contains a series of open-ended requirements which describe the project you’ll be building. There are many possible ways to correctly fulfill all of these requirements, and you should expect to use the internet, Codecademy, and other resources when you encounter a problem that you cannot easily solve.

**Project Goals**

In this project, you’ll be building your own basic design system for a website. In essence, you’ll be building a website to help you build MORE websites in the future! On your site, you’ll collect all the colors, fonts, and some of the repeating styles.

**Setup Instructions**

If you choose to do this project on your computer instead of Codecademy, you can download what you’ll need by clicking the “Download” button below. If you need help setting up your computer, read our [article about setting up a text editor for HTML/CSS development](https://www.codecademy.com/articles/visual-studio-code).

[Download](https://content.codecademy.com/PRO/independent-practice-projects/website-design-system/website-design-system-starting.zip)

**Tasks**

**0/7 Complete**

Mark the tasks as complete by checking them off

**Prerequisites**

**1.**

To complete this project, you should have completed the Codecademy lessons covering the [CSS box model](https://www.codecademy.com/courses/learn-css/lessons/box-model-intro), [display and positioning](https://www.codecademy.com/courses/learn-css/lessons/css-display-positioning), [colors](https://www.codecademy.com/courses/learn-css/lessons/color), and [typography](https://www.codecademy.com/courses/learn-css/lessons/css-typography). You can use more advanced CSS layout concepts like flexbox, CSS grid, or responsive design, but they won’t be required.

**Project Requirements**

**2.**

In this project, you’ll build a style guide/design system for your website. Developers and designers use design systems in order to organize the look and feel of a website or web application. Design systems help to document the visual vocabulary and reusable components of a large website.

As your websites grow larger and more complex, having a consistent set of styles will help them stay cohesive and enjoyable for users! It also helps as you start working with other developers, as you can have a consistent reference for how your website should be built.

Your style guide/design system will include at least a section on colors, fonts, and specific text styles, but it can be extended to as many other areas as you’d like. You can take a look at our [basic example site](https://content.codecademy.com/PRO/independent-practice-projects/website-design-system/example/index.html) for a taste of what you’ll be building (yours will be customized).

**3.**

Your style guide should include a labeled section outlining the colors to be used.

Each color should have:

* A name (you can come up with this name yourself)
* A box displaying the color itself.
* The CSS value used to create the color (in hex, rgb(), rgba(), hsl(), or hsla()).
* An optional description of how you intend to use the colors.

**4.**

Your style should include a labeled section of the fonts to be used (we recommend importing fonts from [Google Fonts](https://fonts.google.com/)).

For each font, the goal is to to demonstrate the various styles that you intend to use (font weight, style, size, and text decorations). To do this, you should include the following for every font:

* The font name
* A series of styles that you intend to use (for example, styles could include: plain, bold and italic, underlined).
* For each style, include a sentence or series of words to demonstrate the text style (our example site uses [the quick brown fox…](https://en.wikipedia.org/wiki/The_quick_brown_fox_jumps_over_the_lazy_dog)).

Make sure to display each of the sections in the proper font, size, and style.

**5.**

In addition to the list of possible fonts, your style guide should include a section for specific text styles for page elements. For example, you could set main heading (<h1>) guidelines about text size, fonts, font styles, font weight, and more.

Include at least 3 element styles, and ideally as many as you’d like to use in future sites.

For each page element, include:

* A page element name (for example, “Subheading”).
* All style rules (for example, font-weight: 700).

**6.**

Add any additional styles that you’d like to include to make your style guide/design system look great. This could include positioning, layout, additional colors, box model properties, and more.

If you’d like some inspiration, you can look to the following samples:

* Salesforce’s [Lightning Design System](https://lightningdesignsystem.com/utilities/text/)
* Google’s [Material Design](https://material.io/design/)
* Twitter’s [Bootstrap](https://getbootstrap.com/).

**Solution and Extensions**

**7.**

Great work! Visit [our forums](https://discuss.codecademy.com/t/build-a-website-style-guide-challenge-project-html-css/462395) to compare your project to our sample solution code. You can also learn how to host your own solution on GitHub so you can share it with other learners! Your solution might look different from ours, and that’s okay! There are multiple ways to solve these projects, and you’ll learn more by seeing others’ code.

If you’d like to extend your site, consider:

* Adding additional components (you could document the various [buttons](https://getbootstrap.com/docs/4.0/components/buttons/) that you’d like to use).
* Learning how and then implementing a [responsive design](https://www.codecademy.com/learn/learn-responsive-design) so that your site looks great on all screen sizes.
* Adding navigation and splitting up your site so that each section has its own HTML page.