

Task-by-Task Guide

If you'd like a little more support while completing this project, explore this step-by-step resource to get additional hints and resources to help you along each task of this project.

Task 0 - Start with a Plan in Mind

Start With A Plan in Mind

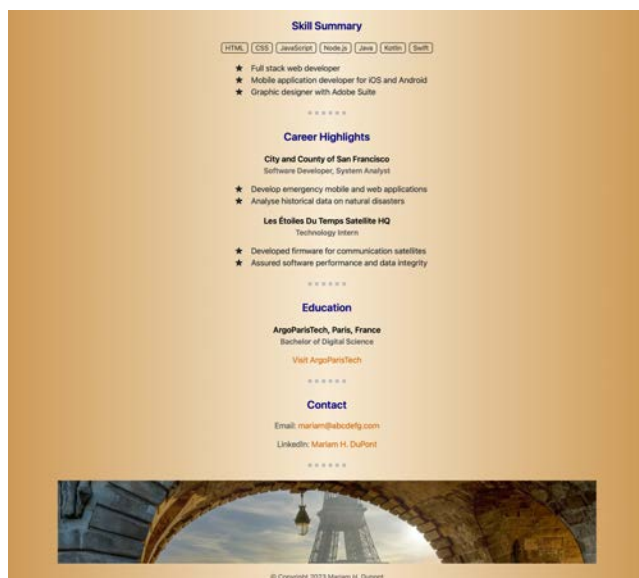
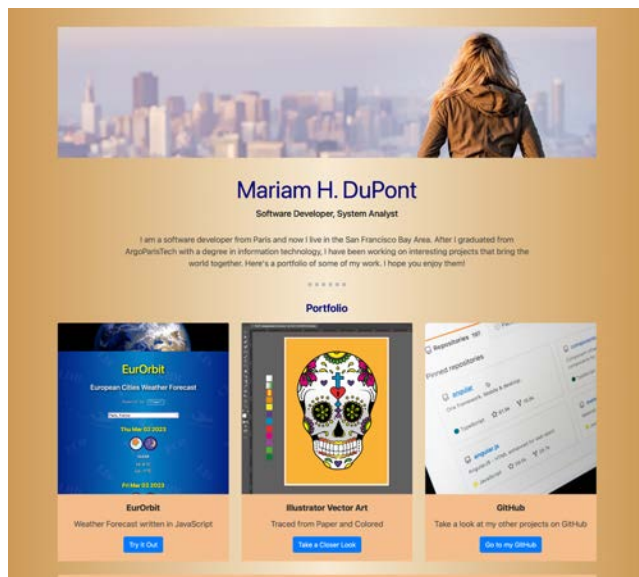
Before you begin, consider taking a step back to plan your steps. Properly planning your project, or scoping, will greatly benefit you; scoping creates structure while requiring you to think through your entire project before you begin. You should start by stating the goals for your project, then gathering the data, and considering the analytical steps required. A proper project scope can be a great road map for your project, but keep in mind that some analyses you start may become dead ends which will require you to adjust your plan.



Task 1 - Explore and Identify Website's Layout and Design

The most important step is deciding on what layout you want to use in your website. Do you want a single page or multiple page design? If you do not have a lot of information to present, a single page design might be a better choice as it is easier to maintain. Go ahead and browse the web for ideas. Some people find it helpful to sketch out the layout on a piece of paper.

In this example we are using a single page design. However, you are free to choose your own layout.



Task 2 - Retrieve and Organize Website Resources

Once you have your layout ideas. Gather information and resources that you want to include in your site. Bear in mind that this will be an on-going process and you might find yourself doing this again in a later step.

These resources might include but are not limited to:

- **Documents:** that you want to allow the viewer to download, for example, your résumé or business card.
- **Social media profiles:** Have a list of what you want to include and make sure the information on them is up-to-date.
- **Websites:** that represent your work that you want to link to.
- **Graphic images and design element:** be sure to use images that you own or are royalty-free. Using graphics of unknown origin can lead to legal actions and undermine your credibility as a designer or developer. There are a lot of websites where you can find these. For example: [Pixabay](#). You can also search for what you need by adding “royalty free” after the subject in any search engine. Some sites require attribution to use their images. Make sure you follow the instructions.

Hint

Avoid including sensitive personal information such as your residential address, personal phone numbers or identification document numbers.

More Resources:

- Royalty-free graphics: [Pixabay](#)
- Free icons (requires attribution): [Flat Icons](#)

Task 3 - Use HTML to Create Website's Outline

Layout the information on each page of your site using HTML only. Some people find it easier to stick with basic tags such as `<h1>` to `<h6>` and `<p>` at the beginning just to get the data on the page. Don't focus on the fine-tuned formatting for now, we will do that in the next step. The idea is to put the data that needs to be displayed on the pages first.

Hint

- Stick to basic tags, don't worry about presentation details
- Put the data that needs to be shown on the page
- Do text first, use image placeholders and then add them after the text is finished

Task 4 - Apply CSS to HTML

Now it is time to fine-tune the presentation of your data with CSS. Gradually apply classes or modify built-in HTML styles to suit your needs. Organize your classes such that changes can be easily made later without unwanted side-effects. In this step, you might find it necessary to fine-tune the HTML that you did in the last step. Also, one of the important concepts of web design is to make your page “responsive”, meaning adapting to different device screen sizes so keep that in mind.

If you are well versed in other CSS frameworks such as Bootstrap. You may opt to use it instead.

Hint

If you are using Visual Studio Code, the Live Server extension allows you to see the effects of your code in real time.

Most modern web browsers (including Google Chrome) have console displays that you can use to debug CSS and JavaScript.

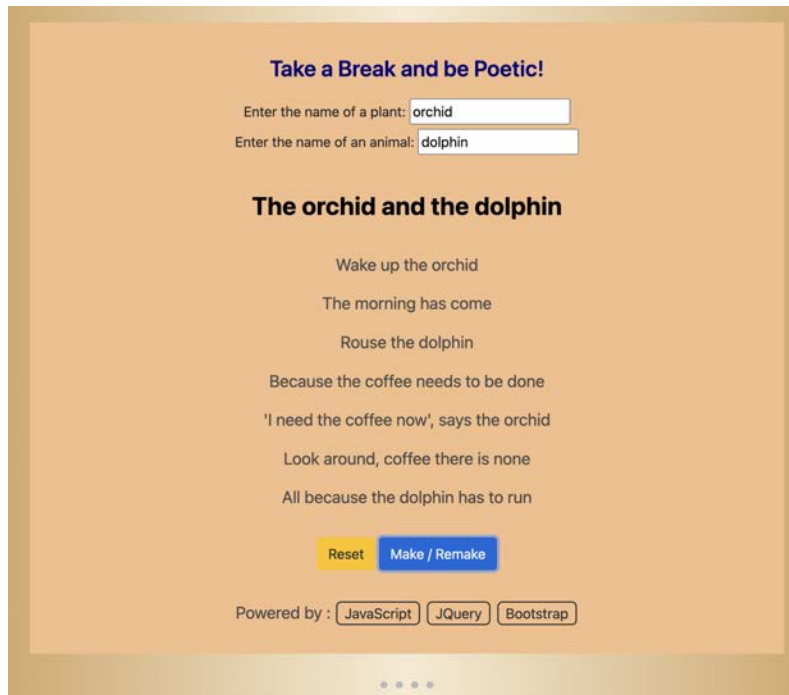
More Resources:

- [Build Fast, Responsive sites with Bootstrap](#)
- [Debugging with the Chrome Browser](#)

Task 5 - Implement Behaviors Using JavaScript

A good website retains user interest by providing interesting visual cues or user interactivity. This will also be your chance to showcase your proficiency in JavaScript. In this example, a simple applet keeps the viewer entertained. Other useful ideas include mini-games and famous quotes.

If you are well versed in other JavaScript frameworks such as JQuery. You may opt to use it instead of plain JavaScript.



Hint

Most modern web browsers (including Google Chrome) have console displays that you can use to debug CSS and JavaScript.

More Resources:

- [Using the JQuery Framework](#)
- [Debugging with the Chrome Browser](#)

Task 6 - Beta Test Website Before Launch

Thoroughly test your website and fine-tune any formatting issues. Make sure all external links work and all your JavaScript work as intended and free of errors. If possible, having someone else test your work might reveal bugs that you won't find otherwise.

Finally, we can wrap up the project. You can write a conclusion about your process and any key findings.

Hint

The main components that you will want to include:

- What did you learn throughout the process?
- Are the results what you expected?
- What are the key findings and takeaways?