

Project 3 Hobby/Organization Site Workflow

1. Pick a topic

The trick to this is picking an idea that you can break out in chunks. For instance, say you like to watch local music. You can break your site out into a homepage describing your site and three of the best venues for local music in San Marcos.

2. Select a Bootstrap theme

It's tempting to choose the most visually appealing template, but consider what works best for the scope of your site. You CAN choose a one-page template with links to different sections, but make sure each section contains a decent amount of content. To understand how in-page linking works, read about it [here](#).

Download the template, move it into your documents, and rename to the title of your hobby site. The above example could be called something like, `local-music`

3. Create a logo or unique element to signify your site

Either replace the logo in your theme or create a new spot for a site logo.

4. Choose a color scheme

If you go to the [Kuler color generator](#) you can upload a photo and generate a color scheme based on that. Establish a primary color in your scheme. Now refer back to your newly created project. Preview it in your browser. Do you see a reoccurring color? Identify that color in your CSS and find and replace that hexcode with the hexcode you chose in the color generator.

Hint for Sublime users: Highlight the old hexcode, hold down `cmd` (Mac) or `ctrl` (PC) and press the `D` key. Keep pressing `D` to see that you're now highlighting each occurrence of that hexcode.

5. Build out your content

This is where you start to gather photos, write out the information, and plan out your navigation. If you need help reconfiguring the nav in your template, refer to [Bootstrap's nav component](#).

6. Integrate your content into your template

This will take some investigative work and some time to familiarize yourself with how this template works. You didn't write the code, but you know the language now. In this step Chrome's Developer tools can be helpful to figure out what content is where in your template.

Start with the basic information. For example, swap out the title tags, contact info, and add the date in the footer. You want to make this template adapt to your content and become unique to your brand. The goal is to use the bones of the template, but add your own personal touch to make it unrecognizable from the original version.

7. Add custom CSS file

Create a new CSS file and save it alongside the other CSS files. You'll then link to the CSS file after the others. You can remove the `.min` from the file name in some cases (if the un-minified file exists) and it's a bit easier to read. It would look something like the following:

```
<!-- Bootstrap Core CSS -->
<link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">

<!-- Theme CSS -->
<link href="css/clean-blog.css" rel="stylesheet">

<!-- My CSS -->
<link href="css/custom-css.css" rel="stylesheet">
```

This will help reduce the clutter of styles in the other files and keep them intact so you don't alter them beyond repair. For the most part, the last rule wins in CSS; however, there are plenty of exceptions. This [handy guide](#) from a CSS whiz Estelle Weyl will help you with understanding battling CSS rules.

8. Publish your site to your new subdomain.

Refer to the folder we created on your servers through Reclaim. Go back to your local working directory and add the contents of that folder to your server.

