

Brian Leip - Portfolio Project 1 - Final Report

Content Creator Website

Link to website: <https://content-creator-site-template.netlify.app/>

Link to github repo: <https://github.com/BrianLeip/content-creator-site-template>

1. Introduction

- This project provides a pre-made “template” website for content creators that can be used to show off their portfolio, courses/tutorials, a store, a contact page, and a newsletter sign up form.

Features include:

- Landing page
- Gradient background
- Smooth scrolling script
- Fixed header with brand name on left and Navbar on right
- Navbar is responsive and becomes a toggler on medium or smaller screens
- Footer with newsletter sign up, social media links, and copyright info
- Large, visually appealing display of portfolio pieces
- Separate page for Courses that customers can sign up for
- Store that allows customers to buy directly from the creator
- Contact page where customers can send the creator a message
- Newsletter sign up / email capture

2. Design and Implementation

Overview

- For this project, I began with a survey of other content creator websites available online, along with some other pre-made templates, and another website that I had previously created in HTML and CSS only.
- From those influences, I noted similarities (clean, simple, large visuals) and added in some other features that I wanted such as a Courses page and a Store page.
- I separated the project into separate logical “groups” and created a fixed navbar with links to those sections.
- At first I started with everything together on 1 page, but as the project grew, I eventually moved each section to separate pages, using the same navbar to link them.
- Formatting was done with a combination of bootstrap classes and custom css classes.
- SCSS / Sass pre-compiler was used since it’s very handy to be able to assign colors to variables and use them throughout the css to create a consistent look.
- For color palette, I referenced a lot of various tools, guides, and recommendations online. After testing out different combos, I ended up with around 10 colors that complement each other well, and give a “calming” effect, with the occasional contrasting color for “pop”.

Responsiveness

- I used a combination of Bootstrap’s responsive classes with custom css media queries.
- This was more finicky than expected, and it took a lot of iterative testing until things “looked right”. But eventually I was able to wrestle everything down.

Navbar

- Fixed navbars make for easy navigation, along with a clear visual of which page you are on, and also the brand, so I went with that.
- Layout is brand name on the left (logo included), and navbar on the right.
- On medium or smaller screens, the individual navbar links become nested within a toggler, and the font size of the branding shrinks.
 - Getting this layout working properly was also tricky to pin down
- The navbar is identical on each page, aside from which link is designated as active.

Footer

- Footer has a large, prominent display for customers to enter their email and sign up for a mailing list.
- The Submit button for the email sign up uses Bootstrap's inline button class
- Under that are social media links, using icon fonts from font awesome.
- And beneath that is the copyright info, in a smaller/lighter font
- The footer is consistent across on all pages.
- *Note - I started attempting to get the email sign up to be functional (send emails to a Google Sheet), but since we haven't covered back-end yet, it didn't pan out. Will circle back to this one later.*

Portfolio Page (index.html)

- I debated between using Bootstrap cards, or a Carousel. I like the look of Cards (especially when they have a nice shadow behind them and subtle rounded edges), so I opted to use cards.

- I may change this in the future though, since I like that carousels auto-display recent projects. But for now the cards turned out nice so I'll stick with that.
- For a visually appealing look, I made the 1st project very large (12 of the 12 bootstrap sections), and projects 2 through 4 smaller size cards so that multiple fit side by side..
- A 3d shadow and rounded edges was added to the cards.
- It seems that most bootstrap cards have the hyperlink on only the picture part of the card and not the full card. I preferred the full card to be the link so I switched it up a bit.

Courses Page (courses.html)

- For this page, I wanted to create separate sections that can contain multiple courses. So I created "Beginner", "Intermediate" and "Advanced" sections.
- A complementary red color was used for the separate sections so that they pop.
- Within each section, I added Bootstrap cards of varying quantities to represent separate courses.
- These cards are much smaller than the portfolio cards so that many can fit on the same row.

Store Page (store.html)

- After realizing that we haven't learned quite enough yet in this course to create a full store page, I added a fun little "Under Construction" message here.
- After we cover what's needed to create a store (either React course or Back-End), I'll finish this part up.

Contact Page (contact.html)

- Created a form section with 4 inputs and a submit button
 - First name

- Last name
- Email address
- Message
- All 4 inputs use HTML5 types, are flagged as required, and have placeholder text.
- I also used a container along with Bootstrap's form-control classes for the inputs.
- As with other parts of the website, this is "front-end" only and not fully functional. I'll connect the front-end to javascript and back end in future portions of the course.

Deployment

- I had heard good things about Netlify and how you can publish a site directly from github, so I wanted to test it out.
- It turned out to be shockingly easy to do.
- Simply sign up, click a few buttons to connect to your github repo, and the page is live!
- Will continue to experiment with Netlify deployment along with tools like static site generators, custom URLs, and https security.

3. Conclusions

Best features of the project:

- I'm actually pretty happy how the overall look and feel turned out. Has a calm, clean, minimalist feel.
- The website is very responsive. Adapts to screen sizes of all types.

- Can be modified relatively easily by a content creator using this as a starting point. Especially since the SCSS colors are assigned to variables, they can be swapped out for a new palette giving the website a different feel.
- The icons from font awesome give it a nice professional looking touch (e.g. the social media links in the footer)
- The multi-page approach turned out to work better than the single page in this case.

Shortcomings of the project:

- Not connected to any functionality (javascript or database)
- Though that is what we expected since the course has only covered front-end so far
- I could potentially add more features and functionality with Bootstrap modals, popovers, etc. But at the same time that may overcomplicate the simple clean look and not add much value.

Final Thoughts

- Overall, it was a fun project that turned out to be trickier than expected but excellent practice.
- With programming, it's surprising how often something *should* work, but it doesn't for inexplicable reasons, and you have to troubleshoot why. The only way to get better and minimize the amount of time troubleshooting and learning weird quirks of each language, is to code, code, code.
- If you keep building, you'll continue learning and get better every time.