



Elizabeth MacCabe

Web Design & Development student at @solentofficial.

May 10 · 13 min read

Creating PatternLibrary from Scratch

PatternLibrary is a responsive front-end pattern library that comprises of various components to build a website.

View PatternLibrary's documentation [here](#) and the GitHub repository [here](#).

<https://medium.com/@eamaccabe/creating-patternlibrary-from-scratch-bf9279f83a34>

PatternLibrary

[Get Started](#)

Components

Welcome to PatternLibrary

PatternLibrary is an easy to use responsive front-end framework and a perfect addition to your website.

Download

PatternLibrary

This is the standard version of PatternLibrary that comes with CSS and JavaScript files. This option requires very little setup so is perfect for if you are not familiar with Sass.

[PatternLibrary](#)

Sass

This is the Sass version of PatternLibrary that comes with all of the source SCSS files as well as the JavaScript file. You will need a Sass compiler if you wish to use this version.

[Sass Version](#)

Introduction

Project Background

A pattern library, often called a framework, is a series of user interface and design elements that are created to be reused throughout a website. Elements within a pattern library typically consist of buttons, slideshows, navigation and carousels. This is a very limited list as a pattern library can be very extensive in its amount of website elements

that it contains. Well known frameworks include; [Bootstrap](#), [Foundation](#) and [Materialize](#). All three of these libraries contain various UI elements that users can apply to their own website.

Project Aims

The aim of this project was to produce a pattern library from scratch using research on pattern libraries and frameworks to understand how they work and how they are created. It was also required that a documentation website was produced alongside the pattern library. This documentation would provide examples of the UI elements in the pattern library, explain to users what they are and how to use them by providing code samples that the user can copy and paste straight into their own website.

• • •

Research

What is a Pattern Library?

Building a design from scratch every time you want to make a website is very time consuming and tedious, not to mention rather boring. In [this Design Lab article](#), it is explained that a pattern library is a solution to this issue by providing a reusable set of components for building a website with ease. Having a pattern library allows you to reuse bits of code throughout a website without having to specify new styles for each element added to a website.

Building a Pattern Library

Using a Tool

The first option for building this pattern library was to use a tool to aid the process of building. [PatternLab.io](#) was considered as it allows pattern libraries to be created with pre-existing user interface elements. This tool is useful for those who are not familiar with how pattern libraries are created and want a simple and fast approach to follow.

From Scratch

The second option for building this pattern library was to make it completely from scratch using my own styling. This approach gives far more control to the user and allows unique pattern libraries to be

designed and produced. By building a pattern library from scratch, you can use your creativity to build elements in whichever way you wish.

The Chosen Option

After becoming more familiar with the Pattern Lab website and documentation, it was clear that I would much rather design and build my own pattern library from scratch. I didn't want to spend more time learning how to use a tool when I already knew all that I needed to build the pattern library that I want to.

Pattern Library Documentation Inspiration

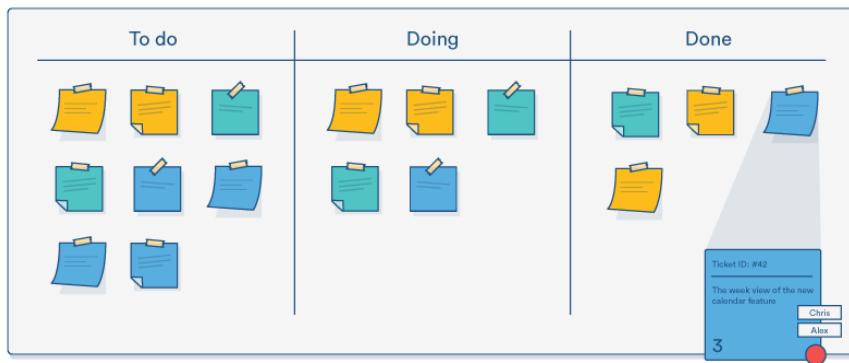
There are hundreds of pattern libraries that already exist online. There were four main pattern libraries, also known as frameworks, that I found which had the best documentation that was easily understandable and visually appealing. These four pattern libraries were; [Bootstrap](#), [Materialize](#), [Bulma](#) and [Skeleton](#). The documentation website of these four pattern libraries to understand how they laid out their pattern library elements, how provided code samples for users to copy and how they provided files for their users to download and use their product.

... .

Design Process Method

[SDLC](#) or the Software Development Life Cycle is a process used to develop software of high quality and in a short amount of time. There are various model types that have been established based on the SDLC process. The [Kanban method](#) is one of the many ways to implement the [Agile methodology](#) which is based on the SDLC process. Although the Kanban method was originally developed for [Toyota's Production System](#), it has now become known as a sign-based scheduling system.

The Kanban method was used throughout this project in conjunction with an online task and project management tool called [Asana](#). Asana provides Kanban-style boards that allow you to input all of the tasks that need to be completed for a project. These boards are typically laid out into three columns; To Do, In Progress and Completed. Tasks are put into their respective columns and are moved around as they are started and completed. This example below shows how a typical Kanban project schedule would be laid out.



The Kanban Tool

By using Asana throughout this project, it provides an easy to use and visually appealing structural system to specify all of the tasks that would need to be completed to meet the criteria for this project. The Kanban style boards provided a quick way to assess each task on its completion and see how the project was progressing by how many tasks were left in the 'Not Started' column.

. . .

Plan and Process of Testing

When building a pattern library it is important to conduct both technical testing and user testing throughout the process of the project.

Technical Testing

Conducting technical testing on the pattern library is important for many reasons. The first and most important reason for testing the pattern library is to ensure that the product that has been produced is of a high standard and all elements function as expected.

Documentation Testing

Additionally, the pattern library will also be tested on the documentation side to check that any code samples provided to the user will actually work when they copy and paste them into their website. This will involve checking that all HTML elements are written correctly without any spelling mistakes and include all of the relevant classes and/or id's to function properly.

Cross-Browser Testing

Browsers often interpret HTML elements, such as buttons and dropdown lists, in different ways especially in Mozilla Firefox. Therefore, by testing the pattern library across all major browsers, it will ensure that all elements work as expected and will be consistent on every browser.

Users of pattern libraries, such as designers and developers, need to know that the product they are going to be downloading and using will be an improvement to their workflow and help them to create a high quality website with ease. A user will expect a fully tested product and, as discussed by Jim Newbery in [this article](#), they should not have to fix any bugs on browsers when they use the elements from the pattern library.

User Testing

Pattern libraries are designed and produced for use by users, most likely designers and developers. By conducting user testing throughout the process of creating this pattern library, it will help to understand the needs of the user and guarantee that they are able to understand how to use the product.

Testing the Pattern Library

User testing will be conducted to test various areas of the pattern library; the first being the UI elements created. A user test will consist of asking participants to interact with these elements and give feedback in terms of design and functionality. Users will provide feedback on things such as the font sizes and colours used and the animations that have been used on elements such as the navigation and buttons.

Testing the Documentation

In addition to gaining feedback from users on their experience with using the elements within the pattern library, feedback will also need to be collected on the documentation. The documentation must clearly explain to users what each element is, what the element does and how they can apply it to their website. Users will provide feedback on things such as if it is clear what the elements do and how it functions and if the code samples provided are understandable.

• • •

Specification

The specification for this pattern library was fairly straightforward and didn't consist of too much as the pattern library at this stage was going to be simple and not include lots of elements.

User Interface Elements

- Grid system using flexbox to provide a responsive pattern library with sections, containers, rows and columns for website structuring
- Navigation bar
- Static header element
- Six different types of button
- Call-to-action jumbotron
- Footer with social media icons
- Modal triggered by a button click

Documentation

- Side navigation bar with a dropdown list for components
- Homepage to explain how to download and set up the pattern library
- Page per element outlined above
- Working examples and explanations of each element
- Code samples for each element for the user to copy and paste

. . .

Design & Planning

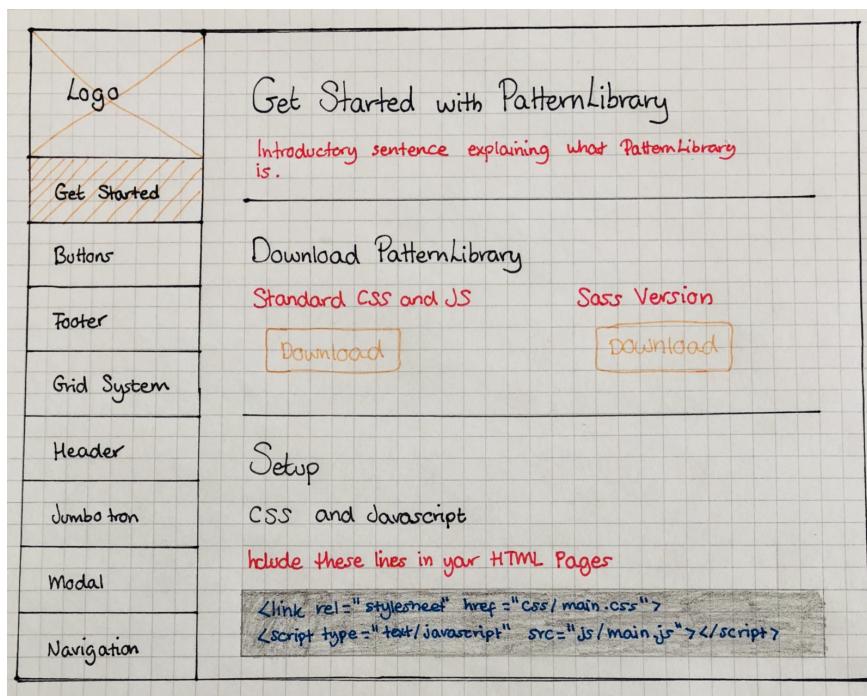
Documentation Designs

Before implementing any part of the pattern library designs for the documentation needed to be made. As the documentation is what the user will follow to use the pattern library, it was essential to design the layout of the documentation. The designs produced showed that each page of the documentation would be consistent with the layout and content and would include:

- A fixed side nav with the logo and each page name.
- The title of the page with an introductory sentence.
- An example of the element on that page with an explanation of what the element is and what it would be used for.
- A code sample underneath the element to show the HTML needed for the user to put that element on their website.

Initial Documentation Sketch

The sketch below shows the basic structure outline that would be present on each page of the documentation for the pattern library. This was the initial sketch designed for the layout and structure of the documentation for the pattern library. As the pattern library doesn't have many components as of yet, it was originally designed to include all pages in the side nav on the left.

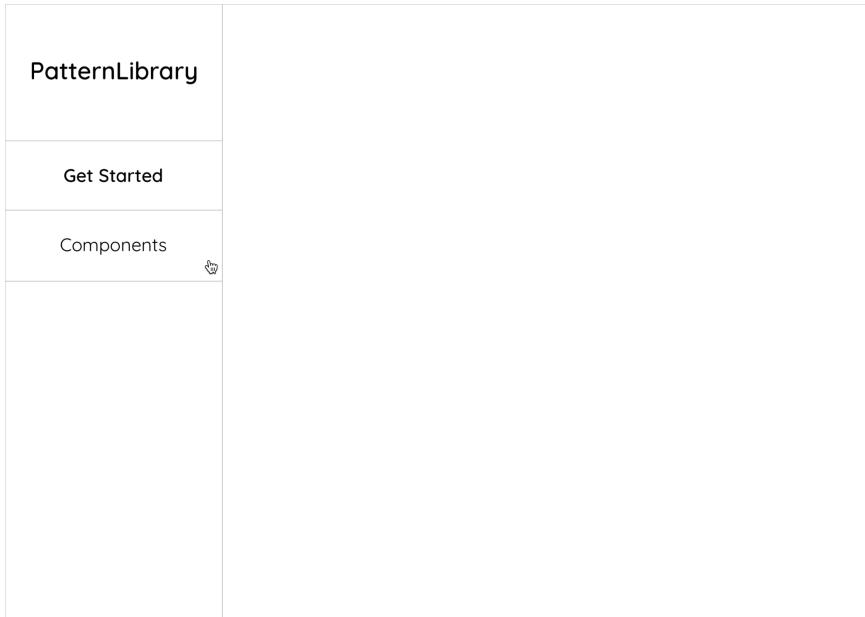


Sketch Design: Documentation 'Get Started' Page

Navigation Improvement

Future development of this pattern library would include the addition of many more website elements and layout options. Therefore, it would make better sense to have a dropdown within the side nav. This side nav could then be updated to include multiple dropdowns for different types of content in the future. The animated GIF below shows a small

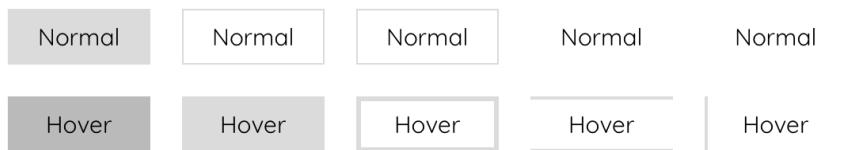
prototype that was made to test how the design would look as a functional object.



Design: Dropdown Prototype Example GIF

UI Element Designs

After designing how the documentation website would be structured and how the content would be presented on it, the UI elements of the pattern library then needed to be designed. The most necessary element to design beforehand were the buttons. As the specification required six different types of buttons to be produced for this project, various types were designed using Sketch to decide which buttons types would be selected and would fit the theme of the pattern library. The screenshot below shows the range of buttons that were designed including how buttons would look when hovered over by the cursor.



Design: Initial Button Ideas

• • •

Experimentation

Grid Using Floats Experiment

Part of the specification outline for this project was the use of a responsive grid for the pattern library using rows and columns to provide web page structure. There were two options for the creation of this grid; the traditional way using floats or using flexbox. Before the creation of this project, I had never used flexbox before and I had very little understanding of how it worked and how to use it.

Therefore, I attempted to build a 12 column responsive grid using the traditional way through the use of floats and percentages for each column. This method took a long time to test out but it worked as desired and provided a 12 column responsive grid which it what was required. As this option had been successful, I decided to begin implementation of the pattern library using this method.

Flexbox Sticky Footer Experiment

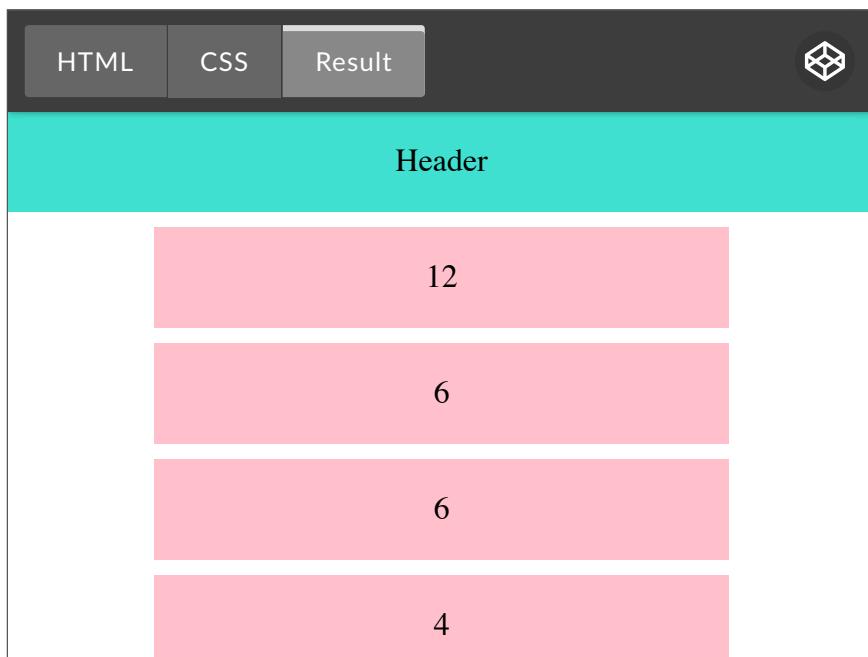
After using the float method for the grid, I ran into an issue when implementing the footer element. The footer would not stay at the bottom of the page if there was little content on the web page. This was a very frustrating problem and, when researching the issue, I found that it was highly recommended to use flexbox to create a ‘sticky footer’ which is a footer that stays at the bottom of the page no matter how little content the page has. The experiment below shows that using flexbox worked exactly how I needed it to.

The screenshot displays a web-based experiment interface. At the top, there is a dark header bar with three tabs: 'HTML' (highlighted in light gray), 'CSS', and 'Result'. To the right of the tabs is a circular icon containing a geometric hexagon-like pattern. Below the header, the page is divided into three horizontal sections: a teal 'Header' section at the top, a white 'Content' section in the middle, and a pink 'Footer' section at the bottom. The 'Content' section is currently empty, illustrating how the footer remains at the bottom of the page even with minimal content.

Experiment: Flexbox Sticky Footer

Flexbox Grid Experiment

As I had just carried out an experiment and achieved the outcome I wanted with the sticky footer using flexbox, it only made sense to scrap the grid I had already created using floats and start over by creating a grid using flexbox. Not only did this technique for creating a grid with flexbox take half the time that the float technique did, it also used far less code and was much easier to understand. Unfortunately, time was wasted by starting implementation using floats as I ran into problems that I would have avoided completely if I had used flexbox from the start. The experiment below shows how I used flexbox to produce a responsive grid and a sticky footer simultaneously with the functionality that I was looking for from the beginning.



Experiment: Incorporating Sticky Footer and Grid Using Flexbox

• • •

Implementation

Implementing Flexbox Grid and Main Structure

After conducting experiments to determine the best way to provide a responsive grid system for this pattern library, implementation of the grid was carried out. Various structural components were defined for the grid and the basic page content including:

- `<header>` , `<main>` and `<footer>` - to contain the respective elements to provide proper structure and ensure the sticky footer works as it should.
- `.section` - full-width sections with top and bottom padding to provide sectional block areas.
- `.container` - limits the element to a maximum of 70% of its parent container. Good for limiting the width of the body section whilst having full-width navigation and footer above and below the main content.
- `.row` - wrapper for columns to be displayed horizontally.
- `.column` - vertical alignment for elements. One `<div>` with the `.column` class is equal to 1 column which is 100% of the parent element. Add more column divs to add more columns.

Implementing PatternLibrary Elements

These are the final user interface elements that I produced for the pattern library. The gifs below show how the user interacts with the elements and show the normal and hover states of all elements that have these features.

Before styling any of the elements, I used HTML to build the basic structure of each element that was going to be created for the pattern library. Once the HTML structure had been figured out, classes were added to the elements to allow them to be styled. I used SCSS or Sass, a CSS pre-processor, to write the styling for each element.

Navigation

The navigation created is a full-width horizontal navigation that, when viewed on mobile devices, becomes a side nav with a hamburger menu icon as the trigger to open and close it. The navigation links were created using a list and hover states were applied to the `` elements to change the background. A `.active` class was also applied to specify the current page by giving it a darker background colour.



UI Element: Horizontal Navigation

Static Header

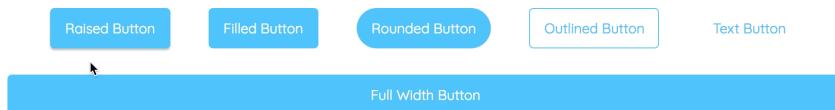
I originally wanted to create a full-width carousel as the header element for this pattern library. However, due to many failed attempts and looming deadlines, I decided to use a static header image instead. The header image uses the CSS properties `width: 100%` and `margin: 0 auto` to keep it centred and fit the entire width of the page and scale to fit all screen sizes.



UI Element: Static Header

Buttons

After designing buttons of many different styles, I picked the six that I thought best match the feel of the other elements in the pattern library and created the best group of buttons as a whole. Certain buttons such as the rounded and the full-width buttons are filled by default but can be styled as outlined buttons using the `.outline` class.



UI Element: Buttons

Jumbotron

The jumbotron is used to provide emphasis on certain elements to draw the users attention to them. A jumbotron can be used for things such as a newsletter signup or a ‘find out more’ section. The jumbotron includes an `<h1>` for the header and `<p>` tags for the rest of the information. An `<input type="text">` field and an `<input type="submit">` button are used for the user to add their details and send the info.

Subscribe to Our Newsletter

Want to keep up to date with our latest news and information?

Enter your email below to join our mailing list. You can opt out at any time.



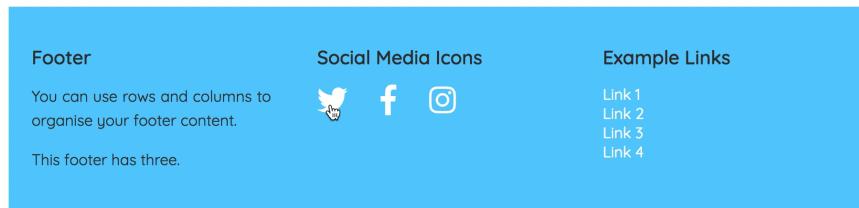
A screenshot of a newsletter sign-up form. It features a light gray header with the text "Subscribe to Our Newsletter". Below this is a white input field with the placeholder "Your email address". To the right of the input field is an orange button with the text "Subscribe Now". A small cursor icon is positioned above the input field, pointing towards it.

 Subscribe Now

UI Element: Newsletter Signup Jumbotron

Footer

The footer uses three `<divs>` with the `.column` class to separate the content but users can have as many columns as they want. The footer can contain lists with links in them to additional pages of the website and uses [Font Awesome](#) to supply the social media icons.



UI Element: Footer with Social Media Links

Modal

The modal is a full screen element that is triggered to open by clicking on a button. Modals can be used to display extra information about something without sending the user to a different page. The modal was positioned using `position: fixed` and margins to keep the modal contents in the centre of the page no matter what size screen it is being viewed on.

The screenshot shows a sidebar titled 'PatternLibrary' with a 'Components' section containing links to Buttons, Footer, Grid System, Header, Jumbotron, Modal (which is highlighted in blue), and Navigation. The main content area is titled 'Modal' and contains a brief description: 'Use modals for dialog boxes, confirmation boxes or other content that can be called on top of the web page.' Below this is an 'Example' section with a button labeled 'Open Modal' and a code snippet showing the HTML structure for a modal.

UI Element: Button Triggered Full Page Modal

Final Product: Pattern Library Documentation Home Page

Conclusion

I asked three participants very simple questions about the pattern library. There were not many questions that needed to be answered.

Questions	Participant 1	Participant 2	Participant 3
Is the pattern library documentation easy to use and understand?	Yes	Yes	Yes
Was it simple to download the pattern library and include the files you needed in your html pages?	Yes	Yes	Yes, except jQuery.
How did you find it adding structure to your pages? Did using rows and columns make sense to you?	Yes	Yes	Yes
Was there anything you didn't like about the pattern library?	No, I liked everything.	No	Would have liked a carousel for the header.

Testing: Feedback from User Testing

As you can see from the questions above, the users found the pattern library mostly easy to use and the documentation was easy to understand. One user did have trouble with getting jQuery to work which was an issue as certain elements that use JavaScript, such as the mobile side nav and the modal, in the pattern library function entirely using jQuery. Aside from that, the participants were happy with one comment mentioning that one user would have preferred a carousel instead of a static image for the header element.