7Factor Corporate Website with JavaScript and Multimedia Assets

A Major Qualifying Project Report

Submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science

In cooperation with 7Factor, LLC

Submitted May 18, 2021

By:

Danya Baron
Molly M. Wirtz
James E. Witt

Project Advisor:

Professor Joshua Cuneo

This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see http://www.wpi.edu/academics/ugradstudies/project-learning.html
Abstract

7Factor, a software company specializing in DevOps and cloud solutions, needed a website that would reflect its values and quality of work. After careful consideration for our technologies, we began to develop a fully functioning React.js website from scratch, implementing the provided UI designs page by page. By the end of the project we were able to complete a responsive, efficient, compatible, and documented website that 7Factor will be able to adapt for future work.
Acknowledgements

We would like to thank the following people for their involvement and help throughout this project:

- Joshua Cuneo, our project advisor
- Jeremy Duvall, CEO of 7Factor
- Tom Bell, 7Factor brand consultant
- Kenneth Jørgensen, 7Factor developer
- Chelsea Green, 7Factor developer
- Ahmed Ismail, 7Factor developer
- Alyssa Tichenor, 7Factor marketing

Without their guidance and feedback, this project would not have been possible.
# Table of Contents

**Abstract** .................................................................................................................................................. 2  
**Acknowledgements** .............................................................................................................................. 3  
**Table of Contents** .................................................................................................................................. 4  
**List of Figures** ...................................................................................................................................... 6  
**List of Tables** ...................................................................................................................................... 7  
**Executive Summary** ............................................................................................................................... 8  
  **Introduction** ......................................................................................................................................... 8  
  **Background** .......................................................................................................................................... 8  
  **Methodology** ....................................................................................................................................... 9  
    **Objectives** .......................................................................................................................................... 9  
    **UI/UX Design - Design Comps** .......................................................................................................... 9  
    **Technologies Research** ..................................................................................................................... 10  
    **Technical Implementation & Code Structure** .................................................................................. 11  
  **Obstacles** .......................................................................................................................................... 12  
  **Results** .............................................................................................................................................. 12  
  **Future Work** ....................................................................................................................................... 13  
**Introduction** ......................................................................................................................................... 14  
**Background** ......................................................................................................................................... 15  
**Methodology** ....................................................................................................................................... 17  
  **Objectives** .......................................................................................................................................... 17  
  **UI/UX Design - Design Comps** .......................................................................................................... 17  
  **Target Audience Evaluation** ............................................................................................................. 18  
  **Technologies Research** ..................................................................................................................... 18  
  **Technologies Decisions** .................................................................................................................... 21  
**Technical Implementation & Code Structure** ..................................................................................... 21  
  **Page Structure** ................................................................................................................................... 21  
  **Header** ............................................................................................................................................... 25  
  **Footer** ............................................................................................................................................... 26  
  **Floating Menu** ................................................................................................................................... 26
List of Figures

Figure 1: 7Factor’s original homepage
Figure 2: 7Factor’s original case studies page
Figure 3: Home page “We Build Good Things” section
Figure 4: Solutions page “Kanban” section
Figure 5: Case Studies page “Our Work” section
Figure 6: Location page “Teams Around the World” section
Figure 7: Work with Us page
Figure 8: Header implementation
Figure 9: Footer implementation
Figure 10: Floating Menu implementation
Figure 11: “Who are We” implementation featuring triangles
Figure 12: A CTA button (left) and a hovered CTA button (right)
Figure 13: The Home carousel
Figure 14: Clubhouse dashboard
Figure 15: Concourse dashboard
List of Tables

Table 1: Collection of original milestones generated at the beginning of the project
Table 2: Collection of final milestones completed by the end of the term
Executive Summary

Introduction

Our sponsor, 7Factor LLC, is a Software as a Service (SaaS) company that uses DevOps, cloud technology, and their business expertise to advise, create, and deploy custom software solutions. Their clients range from large enterprises to small startups, and they pride themselves on producing software of the highest quality. The goal of this MQP was to build an interactive website that showcased 7Factor’s work, mission, and values. With the help of 7Factor and our advisor Joshua Cuneo we were able to complete a fully functioning and responsive React website. In this paper we will describe the obstacles we overcame, our evolving timeline and goals, our final results, and the future work we will hand off to 7Factor developers.

Background

Having a professional and informative website is arguably one of the most important things a business can do; a well-designed website can set a business apart from its competitors and builds trust with its audience by creating an impression of openness and competence (WebFX, Inc.). 7Factor’s current website was created without input from any UI/UX experts, and they believe it is poorly designed, visually unappealing, and unable to incorporate fresh content. As seen in Figure 1, this website is a poor reflection and sales pitch for 7Factor, given their high standards and quality of work.

What 7Factor wanted from their new website was a product that looks polished and professional and reflects the culture and competence of the company as a whole. 7Factor provided detailed design comps to serve as a blueprint for the site, complete with instructions on animations, hover interactions, and more. Pages included in these mockups were a Home page, a Case Studies page,
a Locations page, a Solutions page, and a Work with Us page. Various assets were also provided in image, SVG, and video format, and accompanied by an official brand color guide. Apart from requesting we include some of their DevOps tools, such as Concourse, in our development, we had the freedom to choose whichever languages, frameworks, and technologies we saw fit to create an interactive replica of the design comps.

Methodology

Objectives

Our goal was to develop a feature-rich website for our sponsor per their provided design specifications; to achieve this goal we collected the following list of objectives:

- The website pages, content, styles, and behavior must exactly match that specified in the design comps, unless otherwise specified by 7Factor.
- The website need not be completely responsive, but it should look reasonable on different screen sizes and mobile devices.
- The website must be reasonably backwards and cross compatible across browsers, as well as be acceptably efficient when running / loading.
- The website should be developed using 7Factor’s Continuous Integration/Continuous Deployment (CI/CD) and workflow management systems.
- Our design and technical decisions, development and testing processes, workflow, milestones, and conventions must all be documented and source-controlled, as well as shared with 7Factor.
- The technical implementation of the website should be adaptable enough to allow 7Factor’s developers to extend the website’s functionality (i.e. dynamic content from a dedicated backend) beyond the requirements of the design comps for this project.

UI/UX Design - Design Comps

Before the onset of this project, 7Factor contracted a professional designer to provide design comps for the User Interface (UI) of every page in the website. The design comps included assets in varying sizes/formats, color palettes, and animation specifications per page, all of which we were asked to replicate exactly. The new design for each page sports a hybrid single-page design with both a top menu and a secondary floating menu. The new pages have complex asset layouts and animations that serve content in unique ways while also providing ample space for fresh content and additional pages. These design comps are included as supplementary files to this report.
Technologies Research

Our team had considerable freedom to choose the technologies, frameworks, and implementation of the website, which would consist primarily of a frontend. We decided to briefly research several different solutions to determine the most appropriate approach. When looking at technologies, we considered both the features we would need to implement and the non-technical aspects. The features we considered included single-page site support, a floating menu and header implementation, an HTTP client for Rest API calls, and form validation & handling. The other aspects we considered included popularity, scaling, relative development speed/difficulty, and the learning curve.

Based on our above considerations, we researched and came up with the following technologies as potential options:

- React.js
- Angular.js
- Vue.js
- Pure JavaScript, CSS, HTML
- Flask (Python)
- Typescript

Upon researching the frontend frameworks above, we ultimately decided on React.js based on our prior experience and its well-supported methods to implement the main features in our consideration. As we only needed limited backend support, we determined that the best solution was a Node.js backend with the Create React App toolchain. Created to help developers learn React, and the best way to create single-page sites with React ("Create a New App"), it would ease our learning curve, accomplish a main feature, and make the website incredibly simple to deploy (reducing the overhead for 7Factor).

Technical Implementation & Code Structure

We used web design best practices wherever possible in this project, which included building each page using a 12-grid structure. All the Call to Action (CTA) buttons and several groups of icons on the pages engage the user by changing color when they are hovered over. We used a uniform set of colors taken from 7Factor’s branding guide to style all the text, icons, and buttons for a cohesive, on-brand facade. Each page contains an identical header, footer, and floating menu to navigate the site, and the majority of pages feature a CTA “Work with Us” button to encourage users to contact 7Factor to begin a project.

The styling of this website was a very important factor in its development, especially since 7Factor expressed they wanted the final product to perfectly match the design comps. We started out the styling of the website using pure CSS, but switched to SCSS to match 7Factor’s development practices and create more dynamic styles. To positively affect the website performance, user experience, and SEO rankings we used SVG assets wherever possible and re-seized non-SVG assets.
Workflow

To ensure that our development stayed on track and that we could deliver a finished product by the end of this project, we created a series of milestones to follow. Throughout the course of this project our requirements and priorities evolved, leading to everchanging milestones. Table 1 shows the completed final milestones.

<table>
<thead>
<tr>
<th>Milestone 0</th>
<th>Gather requirements and resources, conduct research, choose technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milestone 1</td>
<td>Setup workflow, environments, and basic backend</td>
</tr>
<tr>
<td>Milestone 2</td>
<td>Implement shared components (header, floating menu, and footer), merge pages into Single-Page Site format</td>
</tr>
<tr>
<td>Milestone 3</td>
<td>Create Home, Solutions, and Work with Us Pages skeletons</td>
</tr>
<tr>
<td>Milestone 4</td>
<td>Implement design comp features in detail (animations, transitions, mouse hovers, etc.) for existing pages</td>
</tr>
<tr>
<td>Milestone 5</td>
<td>Create Location and Case Studies skeletons</td>
</tr>
<tr>
<td>Milestone 6</td>
<td>Implement design comp features in detail (animations, transitions, mouse hovers, etc.) for new pages</td>
</tr>
<tr>
<td>Milestone 7</td>
<td>Apply changes from copies to all pages</td>
</tr>
</tbody>
</table>

Table 1: Final milestones

For our development approach we decided we would use a modified Agile approach, having bi-weekly “stand ups” as a team, weekly meetings with our advisor, Joshua Cuneo, and demos with 7Factor roughly twice a term. We also used 7Factor’s development technologies, including Slack, Clubhouse, trunk-base development with Git, and Concourse.
Obstacles

We faced many obstacles over the course of this project that we had to overcome. First, we discovered in the first few months that the functionality and intention behind the case studies page was ill-defined and ever-changing. Eventually, we agreed with 7Factor that building a Content Management System (CMS) backend for these pages was out of the scope of this project. We also struggled to do work remotely: COVID-19 was still a large factor in conducting our work safely, so along with the logistical and psychological effects of the pandemic, this meant not having access to campus resources and having all communication and collaboration entirely remote. Additionally, 7Factor has employees working remotely in significantly different time zones, and in order to respect that, we had to come up with reasonable meeting times that worked for all parties. Quickly learning new technologies was another obstacle we faced in the beginning of this project. There were many weeks of research, watching tutorials, participating in code reviews, and hands-on learning before we became truly productive working with React, trunk-based development, and Concourse/CI testing. Finally, we faced significant time constraints: we created a lot of objectives for this project, and ended up not developing some due to shifting priorities and limited time. If we had more time, there may have been more objectives that we could have accomplished.

Results

Although some of the lower priority objectives were not completed, we were still able to deliver a website identical to the design comps exactly as 7Factor asked. We poured time and effort into this project to flatten the learning curve of the various technologies, consistently deliver impressive work to 7Factor, and achieve our milestones in a timely manner.

The website pages, content, styles, and behavior must exactly match that specified in the design comps, unless otherwise specified by 7Factor.

We started out with the original designs provided by 7Factor, most of which we were able to fully implement by early C term. However, 7Factor came to us in the middle of C term with some design changes. Due to time constraints, we were unable to complete all of the updated changes that 7Factor gave us, but we were able to develop the higher priority ones.

The website need not be completely responsive, but it should look reasonable on different screen sizes and mobile devices.

We achieved this objective by scaling all the elements in a page by viewport width (VW), aka the width of a user’s browser. This allowed us to shrink and grow a page and its elements responsively by scaling content proportionally, and have the site look good on both mobile screens and extra-large monitors.
The website must be reasonably backwards and cross compatible across browsers, as well as be acceptably efficient when running / loading. To ensure the website would run efficiently and be sufficiently compatible, we followed web design best practices, and tested the website on major web browsers as well as mobile devices so that the site runs smoothly and looks clean, professional, and identical to all users.

The website should be developed using 7Factor’s Continuous Integration/Continuous Deployment (CI/CD) and workflow management systems. We were able to develop the website using 7Factor’s Continuous Integration/Continuous Deployment system, which would test and deploy any code we committed to version control. Learning how to use 7Factor’s workflow management systems was a learning curve, but with their assistance we were able to successfully develop the site in this way, and gain valuable experience working in a DevOps environment.

Our design and technical decisions, development and testing processes, workflow, milestones, and conventions must all be documented and source-controlled, as well as shared with 7Factor. We did most of the documentation for our MQP team within Google Documents, and had bimonthly meetings with 7Factor to show them our progress, discuss our technical decisions and receive feedback. We also used GitHub, Clubhouse, and Slack to properly document our progress and communicate with each other, which allowed 7Factor to monitor our development.

The technical implementation of the website should be adaptable enough to allow 7Factor’s developers to extend the website’s functionality (i.e. dynamic content from a dedicated backend) beyond the requirements of the design comps for this project. When we first spoke with 7Factor regarding the website, they specified that they had hoped for a CMS backend for the website. However, within a month or two of the development of this project, we spoke with 7Factor regarding the backend possibility and said they would rather implement the backend themselves at a later date.

Future Work

Although we completed a great deal of work during this project, it does, as expected, remain unfinished. Several new pages are slated for development by 7Factor employees, including a ‘Meet the Employees’ page, a ‘News’ page, and several individual Case Studies pages, supported by a CMS backend. We are also recommending to 7Factor that creating and implementing a dedicated mobile design and supporting additional accessibility features would significantly improve the user experience for their clients. From a technical standpoint, we are also recommending some amount of formal testing (both unit and integration testing) as well as the eventual migration to React Hooks, both of which were out of the scope of our project.
Introduction

Our sponsor, 7Factor LLC, is a Software as a Service (SaaS) company that advises, creates, and deploys custom software solutions. Using business-to-business (B2B) practices, 7Factor consults with customers to obtain a clear idea of their software needs, and utilizes DevOps and cloud technology to build the final product. They cater to a multitude of businesses ranging from large enterprises to small startups, and they pride themselves on producing software of the highest quality.

The goal of this MQP was to build an interactive website that showcased 7Factor’s work, mission, and values. We created a list of objectives for ourselves that were tied in with our timeline and allowed us to incrementally accomplish goals for the website. The objectives included setting up the website design so it exactly matches the design comps, running the website without the interference of technical bugs, and developing within 7Factors’s workflow management systems. Following these objectives helped us accomplish our tasks in a timely manner while avoiding careless programming. Paired with our projected milestones, we formed a guideline for us to develop efficiently and effectively.

This paper will go through the various steps our MQP team took in order to develop our website. With the help of 7Factor, our advisor Joshua Cuneo, the provided design comps, and our objectives, we were able to complete a functioning React website. There were a few obstacles that we ran into, such as the learning curve of learning React and web design best practices. Our timeline of this project focused on basic requirements gathering and project setup in the beginning, then developing the website’s shared components before moving on to developing complete pages for the website structure. With this timeline we were able to successfully develop a fully functioning responsive website in 21 weeks.
Background

Having a professional and informative website is arguably one of the most important things a business can do. Often referred to as the “digital face of a business”, websites are frequently the first impression a customer has of a company, and can influence perception of everything from branding to customer service. A well-designed and intuitive website can set a business apart from its competitors, and builds trust with its audience by creating an impression of openness and competence (2). Websites that are confusing to navigate, boring, or overwhelming are unlikely to retain the customer, and are bound to have a negative impact on the business.

7Factor has found their current website to be unsatisfactory. As a basic “wireframe” website created by developers without input from any UI/UX experts, they believe it is poorly designed, visually unappealing, unable to incorporate fresh content, and overall very basic. Screenshots of their site are pictured below in Figures 1 and 2, and appear bland and unengaging. As a whole, the website is a poor reflection of 7Factor.

![Figure 1: 7Factor’s original homepage](image-url)
What 7Factor wants from their new website is a product that not only looks polished and professional, but reflects the culture and competence of the company as a whole. Specific pages they wanted were a Home page that really shines to hook viewers in, a Case Studies page to show customers proof of the work they do, a Solutions page to explain the solutions they offer, a Location page to show more about their teams, and a Work with Us page for potential clients to contact the company through were also included in their designs. 7Factor has seven ideologies which they believe “help build human-centric, high performing engineering teams” (7Factor, LLC) and include placing high importance on constant improvement, equality, curiosity, and having a passion for building innovative, elegant solutions. 7Factor stressed that these core values and their overall brand should permeate through the website, expressing to clients that they are an organization that puts clients first while producing software of the highest quality.

7Factor provided detailed design comps to serve as a blueprint for the site, complete with instructions on animations, hover interactions, and more. Various assets were also provided in raw image, SVG, and video format, accompanied by an official brand color guide. Apart from requesting that we use their DevOps environment as part of our development, we had the freedom to choose whichever languages, frameworks, and technologies we saw fit to create an interactive replica of the design comps.
Methodology

Objectives

Our goal was to develop a feature-rich website for our sponsor per their provided design specifications, complete with an adaptable codebase so that they may build off of our work in the future. We produced the following list of requirements in collaboration with 7Factor to evaluate our achievement of this goal, in no particular order:

- The website pages, content, styles, and behavior must exactly match that specified in the design comps, unless otherwise specified by 7Factor.
- The website need not be completely responsive, but it should look reasonable on different screen sizes and mobile devices.
- The website must be reasonably backwards and cross compatible across browsers, as well as be acceptably efficient when running / loading.
- The website should be developed using 7Factor’s Continuous Integration/Continuous Deployment (CI/CD) and workflow management systems.
- Our design and technical decisions, development and testing processes, workflow, milestones, and conventions must all be documented and source-controlled, as well as shared with 7Factor.
- The technical implementation of the website should be adaptable enough to allow 7Factor’s developers to extend the website’s functionality (i.e. dynamic content from a dedicated backend) beyond the requirements of the design comps for this project.

UI/UX Design - Design Comps

Before the onset of this project, 7Factor contracted a professional designer to provide design comps for the User Interface (UI) of every page in the website. The design comps included assets in varying sizes/formats, color palettes, and animation specifications per page.

One of our first steps toward building a timeline for this project was to ask our stakeholders how closely we should follow these design comps. There was a possibility that they disliked one thing or another about the comps, or that they would like to grant us more freedom in designing the website. Both prospects could heavily impact our timeline. However, our stakeholders expressed their wish for us to follow the comps to the finest detail, unless they said otherwise (as stated in our objectives). This requirement greatly reduced the complexity of our timeline, as it would be focused solely on our technical implementation and code.
7Factor clearly worked closely with their designer on the design comps. They expressed to us that they particularly disliked their original site due to how basic and inflexible it was in design, and they feared it was not representative of the quality of their own work. The prospective website sports a hybrid single-page design with a top menu and a secondary floating menu. Each item in the top menu redirects the user to a different page, but most pages feature the secondary floating menu whose items lead to anchors across the vertical length of the page. This system with two menus makes the website feel much more dynamic in contrast to their original site. In addition, most pages have complex asset layouts and animations that serve content in unique ways, which they found to be much more representative of the quality of their work. This new design also provides ample space for fresh content and additional pages. All page content is static in the comps, though 7Factor noted that they will implement dynamic content themselves after this project. We provide the complete design comps as supplementary files attached to this report.

Target Audience Evaluation

Since our team was provided with highly detailed design comps and instructed to strictly follow them, it was not necessary for us to conduct any form of target audience evaluation ourselves. Unfortunately, 7Factor does not know whether their designer conducted any such evaluation either. However, 7Factor did provide us with some details regarding their target audience for the site. Since they provide B2B solutions, they likely expect most users on the site to be company representatives/POCs (Points of Contact), or small-business owners. Thus, in general the user may be expected to be comfortable with the dynamic and complex designs in the comps, though their design would not leave less tech-savvy unattended. A Call to Action (CTA) button leading to their contact page is always on-screen. Their contact page (called the Work with Us page) departs from their dynamic and complex themes with a simple static form, sporting verbose prompts and open-ended text entries. As such, even a less technically-inclined user may easily get in contact with 7Factor. Highlights from the Work with Us page and the rest of the design comps can be found as supplementary files attached to this report.

Technologies Research

Our team had considerable freedom to choose the technologies, frameworks, and implementation of the website. The design comps did not imply the need for a dedicated backend, so we focused on frontend technologies. We decided to briefly research several different solutions to determine the most simple, helpful and appropriate approach(es) to implement the website’s features and fulfill all our goals. We also considered our collective prior experience and comfort level with each technology. Though the fine details of our research are outside the scope of this paper, we provide a summary of our considerations and findings. The following is a list of each item of consideration, in no particular order:
• **“Single-Page Site” Support.** Though not strictly necessary by the design comps, the general website layout and UI/UX suggests a Single-Page site implementation, where the entire site is composed of a single ‘page’ whose content dynamically changes as the user navigates the website (“Single Page Vs Multi-Page Websites: Design Battle!”). This approach resonated with our team, thus we chose to make it a priority early on.

• **Floating Menu Implementation.** Nearly every page on the site employs a fixed-position side menu for user navigation, and we needed the supporting technology to easily make this feature.

• **Header Implementation.** As with most websites, this site requires a static top header present on all pages, and so this was an essential feature to be implemented.

• **HTTP Client (Rest API calls).** Since it was possible that some pages might require custom web API calls, we looked into whether each technology had expedited means of submitting Rest API calls, or whether they were open to pure JavaScript means.

• **Form Validation & Handling.** The Work with Us page in the design comps features a custom form, which is expected to be fully functional. Thus, we looked into the capacity for each technology to handle HTML Form validation within available frameworks.

• **Popularity.** The more popular the technology, the more frameworks, packages, documentation, and community support we can expect to be available. Our team believed this could help us overcome technical challenges and implementation decisions during development.

• **Scaling.** Since future expandability is part of our objectives, we made sure to examine how each technology performs with different project sizes.

• **Relative Development Speed/Difficulty.** We considered how much faster it seemed to generally develop in each technology compared to the others.

• **Learning Curve.** We considered our collective prior experience and which technology would maximize our use thereof.

As we progressed in our research we made some decisions along the way. Though we considered the possibility of working with many different backend solutions for web development, the lack of need for a backend prompted us to choose the simplest option, and so we turned our focus to the frontend. Based on our collective prior experience, our discussion of technologies became focused on the following JavaScript front-end frameworks:

• **React.js.** An open-source JavaScript library for building UIs and UI components. Tens of thousands of websites are written in React, which is developed and maintained by Facebook, with widely available community support. A single-page site design with a static header and footer can be easily achieved using React Routers and the Create React App toolchain. React supports jQuery and Axios for Rest API calls. Most of our remaining features of interest can be satisfied by the Material UI framework (React).
- **Angular.js.** A JavaScript-based framework for building rich web and mobile apps. Tens of thousands of developers use Angular (AngularJS). There might be ample community support, but we found documentation to be somewhat limited. This could induce a greater learning curve. Otherwise, Angular has native support for single-page sites, and handling a static header and footer is fairly straightforward. Angular also handles Rest API calls and form validation on its own (AngularJS). Overall, Angular could be the framework that poses the least technical risk out of those we reviewed.

- **Vue.js.** Another open-source JavaScript library, but for building progressive web apps (Vue.js). Unfortunately, Vue does not have nearly as much market share as Angular or React and it is the newest of the three frameworks, so community support may be slightly sparier (Xie). However, they have very good documentation. Though various third-party packages are available for most of our features, they do not seem to have built-in solutions for most of our features. As such, Vue would likely be the highest technical risk option out of the three.

Other considerations:

- **Pure JavaScript, CSS, HTML.** We considered foregoing any major framework and writing a traditional JavaScript, CSS, and HTML site. Some potential advantages to this route would be a simple learning curve and more freedom in the technical implementation of the site. However, they would come at the expense of potentially higher technical risk (as we would have to resort to more special-purpose third-party packages) and higher development time. We felt that not using a framework was not worth incurring potential technical risk and debt, and decided to use a JavaScript framework.

- **Flask (Python).** We also considered completely foregoing JavaScript. However, this would have presented a much larger learning curve for most of our team in particular, as many of us were unfamiliar with using Python for web development. As such we decided against Python early on.

- **Typescript.** Lastly, we considered using TypeScript over standard JavaScript. Strong typing in code can lead to more robust codebases in general, and better-documented code. TypeScript simply compiles down to JavaScript, so this would not have necessarily replaced our choice of framework. We decided against it due to our time limitation and the additional learning curve it would incur. Since team members would have to learn both the framework and TypeScript at the same time, we anticipated things getting somewhat confusing. Additionally, most community support is provided in standard JavaScript as opposed to TypeScript.
Technologies Decisions

Upon researching the frontend frameworks above, we concluded that any would be appropriate to implement our website. Each has well-supported methods to implement the main features we needed. With this in mind, we opted to use React.js as this is the framework with which our team had the most prior experience.

As mentioned above, our choices regarding backend technologies were almost entirely based on expediting our development process. We determined that the fastest backend solution to implement, given our selected frontend framework, was a Node.js backend with Create React App. Create React App is a simple toolchain created to help developers learn React, and the official React documentation regards it as the best way to create a single-page site with React (“Create a New App”). Not only would it help our members unfamiliar with React, it would also help us accomplish one of our main features in consideration. Additionally, since Create React App is based on the Babel and Webpack toolchains, it would also greatly aid us with cross-browser compatibility, backwards compatibility, and deployment. Another point of consideration is integration of our website with 7Factor’s existing backend systems. Node.js and Create React App make websites incredibly simple to deploy, reducing the overhead for 7Factor to add our site to their support systems.

Technical Implementation & Code Structure

Page Structure

We used grid containers in order to style every page in a 12-grid structure. Grids, particularly 12-grid designs, help make the website more reactive due to each grid being able to scale in proportion to other grid elements, and is considered a web design best practice (Tang). All the CTA buttons and several groups of icons on the pages change color when they are hovered over, indicating a clickable element to reveal different elements or added functionality. We used a uniform set of colors taken from 7Factor’s branding guide to style all the text, icons, and buttons for a cohesive, on-brand facade. Each page contains an identical header, footer, and a sticky floating menu on the right side to navigate the page, and every page except the home page features a CTA “Work with Us” button to encourage contacting 7Factor to begin a project.

The Home page is the first page the users land on, and gives the user a basic introduction to 7Factor, such as who they are and what they do. The We Build Good Things hooks the user in with the large animated logo and the Featured Project button, as shown in Figure 3. The About Us section is used to communicate 7Factor’s mission statement and display the human-centric values of the company to potential clients. The video in the View section (not included in the provided assets
and therefore unimplemented in our project) shows how 7Factor’s employees operate on a day-to-day basis. Below that, the Who Are We section shows 7Factor’s seven pillars in an interactive display of clickable icons and alternating triangle images. Another opportunity to contact 7Factor is presented in the “Help” section which prominently displays a CTA button linking to the Work with Us page. Finally, a carousel containing client testimonials automatically scrolls across the bottom, providing further assurance to potential clients of 7Factor’s skill and capability.

![We Build Good Things](image)

**Figure 3: Home page “We Build Good Things” section**

The Solutions page is used to show the user what 7Factor has to offer, such as their consulting and development services. The How We Build Things section outlines the resources they provide to clients with a description of each below the corresponding title and icon. Below this is a single client testimonial to once again reinforce the skill of 7Factor. The Tech Stacks section, complete with a customized carousel, shows all the different technology that 7Factor leverages to provide the client with a robust product. Figure 4 shows The Kanban section, which deviates from the triangle theme by using boxes of different sizes, colors, and images paired with informative text explaining exactly how they create their solutions.
The Case Studies page showcases some of the projects 7Factor has worked on in the past. The top section echoes the home page with its animated logo, while the Case Studies section highlights key stories where 7Factor excelled in their product creation and client services. Each row in this section, shown in Figure 5, contains a featured study containing appropriate images, the client, and a description of both. Although it is outside of the scope of this project, each of these sections will lead to a page where that study will be reviewed in-depth, and this page will be backed by a Content Management System (CMS) backend so that 7Factor can quickly and easily add new content. Finally, the Past Clients section highlights 7Factor’s most notable clients, demonstrating industry experience and the support of both big-name companies and local small businesses.
The Locations page displays the different locations 7Factor operates in, and the staff at each office. The Teams Around the World section (Figure 6) features a large map of the world composed of triangles, with each location highlighted in orange. A Work with Us CTA button provides another opportunity for potential clients to contact 7Factor to begin a project. Below this are several rows, each containing a triangle image of the city, an address, a phone number, and a link to an Our Team page to show that the locations are more than just buildings; there are full teams and real people working there.
The Work with Us page, displayed in Figure 7, is a simple contact form that allows potential clients to fill in their company, name, email, phone number, which 7Factor location is closest, and an overview of what they want from 7Factor. This is arguably the most important page of the website: every page and the header contains a CTA button to route here, and each page makes a compelling argument to why a visitor to this site should contact 7Factor.

Figure 7: Work with Us page

Header

Our header is where our React Router and Switch live. From the header, the user can navigate to the home page by either clicking on the 7Factor logo in the top left corner or “Home” in the menu. “Solutions”, “Case Studies”, and “Location” all direct to the pages of that name, while the “Start a project with us” CTA button in the top right corner directs to the Work with Us page. As seen in Figure 8, the header menu is customized so that the page the user is currently on is underlined, and each menu item changes color on hover.

Figure 8: Header implementation
Footer

When users scroll down to the bottom of the website they will find the footer, which contains the technologies 7Factor specializes in, and social media links. A picture of the footer is displayed below in Figure 9. Our technical implementation used CSS in order to properly align the rows and columns and exactly mirror the design comps. For the social media icons, we used the assets provided by 7Factor which change color to orange when hovered on. Also, on mouseover the text underneath all of the options of technologies become underlined. The 7Factor logos were provided in the assets that were given to us by 7Factor. Our footer was very simple in terms of development, but still needed to appear clean and informative, as it is displayed on every page of the website.

Floating Menu

The floating menu (Figure 10) is customized to work with any webpage in our project, and can jump to a theoretically infinite number of points. To utilize the floating menu, anchor points have been placed throughout the page with unique “name” properties. These anchor point names are then passed into the floating menu when it is called in. This component creates a button for every anchor point that is passed in, which can be clicked in order to jump to the anchor point tied to that specific button and change its style accordingly. For cases where the buttons are not clicked, the floating menu also styles the buttons based on where the user is on the page. There are also four social media icons always under these jump buttons.
Triangles

Triangles make up a large part of 7Factor’s logo, and therefore heavily influence their website’s design. Although triangles in the design comps appear in many forms, including different colors, sizes, background images, and directions, their relationship to the logo and each other provides an important continuity for the site. Other notable aspects of the triangles were rounded corners, optional text overlay, and the ability to be hoverable. We decided it would be advantageous to create a Triangle component that could be customized within its own file and placed wherever in the page it was needed.

After several attempts of coming up with a robust solution, we settled on using an HTML canvas element, which creates a set space where lines can be drawn in JavaScript to create a custom shape. This was an acceptable solution for us, as these shapes are able to be customized with rounded corners and background images, as well as have support for different sizes. We were able to indicate the custom specifications for each triangle by passing React props to denote the size, direction, text, and color or image the triangle should display.

Although we used these triangle components in development for some months, we eventually ran into some issues. While these components appeared identical to the design comps, when implementing hoverability on the “Who We Are” section on the home page (Figure 11) it eventually became apparent that it was imperfect and prone to bugs, and we pivoted to changing
the structure on the click of a nearby icon instead. This was a more robust method, and more ideal as there is no need to create a different implementation on mobile, on which hover interactions are not possible. Soon after, 7Factor supplied us with updated assets, which included the triangles they desired in image format, already styled with the correct backgrounds and overlays. With no more need for hoverable interactions on specific portions of the triangles, we decided that to use images would be less “buggy”, take less computational effort on behalf of the website, and be easier to make reactive later on.

Figure 11: “Who are We” implementation featuring triangles

CTA Button
Throughout the website are various interactive Call to Action (CTA) buttons to direct users act, whether that be to learn more about the company’s successful work by navigating to a case studies page, or by navigating to the Work with Us page to contact 7Factor. These buttons are either colored or outlined (which invert when hovered), and optionally have an icon placed within the button. Figure 12 shows an example of a CTA button on the Solutions page, both as it appears normally (left) and when hovered on (right). The element of interactivity increases the chances of the user engaging, therefore accomplishing the design goal of the website.
Carousels

The comps indicated the need for two carousels: one on the Home page and one on the Solutions page. The Home page carousel, as shown in Figure 13, holds three different colored slides, each displaying a different client image, testimonial, and logo. The center slide is prominently displayed, while the two secondary slides are set further back. There are triangular buttons on each side of the carousel so the users can navigate them manually, but it also has automated movement. Movement between the slides is seamless as slides appear and disappear in opacity transitions. Although built using the same component, the Solutions carousel is customized differently to match the design comps. It holds 7Factor’s client logos, displaying six at a time, and uses custom arrow buttons to manually move it.

Styling

The styling of this website was a very important factor in its development. Since 7Factor expressed they wanted the design of the website to perfectly match the design comps, there were many steps in getting the styling correct. 7Factor provided us with the proper colors, buttons, triangles, icons, and logos we needed to fulfill their design comps. We used SVG assets wherever possible; as vector graphics rather than pixel-based images, these assets could be resized without compromising quality (Girard).
We started out the styling of the website using pure CSS, where we were able to create effects such as hovering on buttons, highlighting text, and setting the correct hex value colors for each property. We used CSS for a good chunk of our first semester until 7Factor communicated to us that they would primarily use SCSS in their development, and would prefer we implement SCSS over CSS in this site. We then converted all of our CSS files to SCSS stylesheets to match 7Factor’s development practices. Using SCSS allowed us to create global variables in our program with the hex values of 7Factor’s branding colors so it would allow for easier development.

One of the challenges in styling was resizing non-SVG image and icon assets so they were properly aligned with the other content in the website. The image assets that were provided to us were scaled very large, and if added as is, would negatively affect the website performance, user experience, and SEO rankings (Vita). Finding ways to shrink them without compromising the image quality was at first difficult to figure out; what worked for us was resizing them on our local computers, and making the necessary adjustments in CSS/SCSS.

**Workflow**

To ensure that our project stayed on track and that we could deliver a finished product by the end of this project, we created a series of milestones to follow. The initial milestones, created in the first few weeks, are shown below in Table 1.

<table>
<thead>
<tr>
<th>Milestone 0</th>
<th>Gather requirements and resources, conduct research, choose technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milestone 1</td>
<td>Setup workflow, environments, and basic backend</td>
</tr>
<tr>
<td>Milestone 2</td>
<td>Implement header, floating menu, and footer, merge pages into Single-Page Site format, create contact form with backend support</td>
</tr>
<tr>
<td>Milestone 3</td>
<td>Create Home, Solutions, Location, and Work With Us Pages</td>
</tr>
<tr>
<td>Milestone 4</td>
<td>Implement design comp features in detail (animations, transitions, mouse hovers, etc.)</td>
</tr>
<tr>
<td>Milestone 5</td>
<td>Create Case Studies and View Case Study pages, with backend support</td>
</tr>
</tbody>
</table>

*Table 1: Original milestones*
Soon after creating the shared components of milestone two (header, menu, and footer), we decided for the sake of progress to adjust our milestones. 7Factor indicated shortly after the start of the project that some backend work may be needed for the contact form and case studies, but that such work was low-priority relative to the frontend work. Therefore, the contact form was deprioritized, and we moved on to completing the full pages in milestone three. Keeping in mind that 7Factor had requested a few complete pages over many semi-complete pages, we went on to implement milestones three and four simultaneously to produce thorough replicas of their design comps, complete with the animations. After various discussions with 7Factor, our milestones were adjusted again when it was decided that there would be no backend support for the Case Studies after all, and that the contact form should only have enough functionality to connect to their HubSpot.

By our final term we had faced enough changes in scope and prioritization that we created a new set of milestones, shown below in Table 2. One of the more significant changes was splitting some pages into different milestones. The Home, Solutions, and Work with Us pages were completed fully during A through B term, while Location and Case Studies were completed in C term. Splitting the pages this way, instead of completing all skeletons and then adding details, allowed us to quickly develop manageable amounts while adhering to 7Factor’s request of complete pages over half-done work. In the last weeks of our project 7Factor expressed they wished us to focus on finalizing our existing pages according to their newly created final “copies" rather than develop any more pages or features; we therefore eliminated the future View Case Studies page along with backend support for the Work with Us page in favor of these updates.

<table>
<thead>
<tr>
<th>Milestone 0</th>
<th>Gather requirements and resources, conduct research, choose technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milestone 1</td>
<td>Setup workflow, environments, and basic backend</td>
</tr>
<tr>
<td>Milestone 2</td>
<td>Implement shared components (header, floating menu, and footer), merge pages into Single-Page Site format</td>
</tr>
<tr>
<td>Milestone 3</td>
<td>Create Home, Solutions, and Work With Us Pages skeletons</td>
</tr>
<tr>
<td>Milestone 4</td>
<td>Implement design comp features in detail (animations, transitions, mouse hovers, etc.) for existing pages</td>
</tr>
</tbody>
</table>
For our development approach we decided we would use a modified Agile approach. As a team we had bi-weekly “standups” to share our progress, discuss blockers, and assign each other new work. We also shared our weekly progress with our advisor, Joshua Cuneo, and held demos with 7Factor roughly twice a term. We used Slack for inter-team communication, as well as contact with our advisor and client, and Clubhouse for a Kanban-style issue tracker. The first few weeks we participated in code reviews to get everyone comfortable with React, and every month we assigned a project “leader” to take point within the team and with communication with the client.

In addition to Clubhouse issue tracking, 7Factor also requested we use a few other technologies in our development. When using Git version control, 7Factor stated they prefer we used trunk-based development and Concourse, a CI/CD tool that automatically ran a stub unit test on any merge with master before deploying the changes to the hosted site at an AWS S3 instance. A screenshot of our Clubhouse dashboard is shown below in Figure 14. Tasks were assigned under “Stories”, along with a description, and owner, a due date, and several other details.

![Clubhouse dashboard](image.png)

**Figure 14: Clubhouse dashboard**
On a day-to-day basis, we would work independently on developing whatever features or pages we assigned amongst ourselves in our bi-weekly meetings. We would complete these tasks in small chunks, committing and opening pull requests to the main repository of our Git version control often as per the trunk-based development philosophy. Once a pull request was opened, we requested each member of our MQP team, as well as one 7Factor developer, to review the new code. Comments, if any, were quickly made and fixed, and pull requests were merged in a timely manner. Once a pull request was merged into the main branch, the automated Concourse pipeline was triggered to test and deploy the new changes, as described above. It was our responsibility to monitor this pipeline to ensure that the unit test passed with the new changes, something we observed from the Concourse dashboard in Figure 15.

Figure 15: Concourse dashboard
Obstacles

Case Studies Page

We faced additional obstacles when it came to the case studies page in the form of unspecified functionality and potential scope creep. 7Factor had originally hinted at needing a backend for the case studies page, specifically to support a CMS backend that would add, remove, and manage different case studies displayed on the website. After much research and discussion with 7Factor, it was decided that implementing a backend would be outside the scope of this project, and that it would be best for us to create a single static Case Studies page.

Remote Work

This MQP was completed over the course of A, B, and C term during the 2020/2021 academic year. This project was completed remotely, since 7Factor’s headquarters were located in Atlanta, Georgia and while COVID-19 was still a large factor in conducting our work safely, we would have had to work remotely due to 7Factor’s location. Working remotely comes with some obstacles that were prevalent in the development of this project. We had limited access to campus resources that may have been helpful throughout our project, such as the ability to communicate with others that may have known about React and other technologies that we used for this project. Remote communication was also a challenge, especially when finding time to meet with 7Factor. One of 7Factor’s team members, Kenn, was in Japan which was a 14-hour time difference. We had to come up with reasonable meeting times that worked for all parties. We were lucky to have a supportive and understanding team and advisor, but the logistical and psychological effects of the pandemic and related events were factors in this project.

Learning React & Technologies

In order to develop the website, all of us needed to learn React and other technologies rather quickly. Our fellow team member, Lucas, had experience with React, but the majority of us had never used it before. This was an obstacle for the team, because only one of us had minimal React experience. There were many weeks of researching and watching tutorials in order to learn React and apply it to our project. Lacking experience in React presented obstacles for ourselves as we had to constantly learn and adapt to new problems. As part of the solution to this obstacle, the first few weeks we participated in code reviews amongst ourselves to help each other with basic structure, syntax, and elements of React. Other technologies we had to get familiar with were some of 7Factor’s development tools, such as trunk-based development, and Concourse/CI testing. In order to properly deploy the website, we had to learn Concourse, and also actively update our
GitHub branches and merge them frequently, as per trunk-based development principles. Learning all of these technologies and techniques was essential in developing this website.

Time Constraints

Any software development project naturally has many changes in course, and one of the biggest factors in software development is time. We created a lot of objectives for this project, and we ended up not developing a few of them due to both time constraints and shifting priorities. New issues and bugs arose during the development of our project, and as being full-time college students with classes, achieving all of the objectives we set out to do was not possible. Our schedules also allowed us to meet twice a week where we would discuss possible issues that arose during the development of this project. If we had more time, there would have been more objectives that we could have accomplished, such as a functioning backend for the website.
Results

As stated earlier, we had quite a few objectives that we wanted to accomplish during the development of the 7Factor website, and unfortunately not all of them were completed. As with any software development project, additional tasks and new priorities were given by 7Factor, causing us to occasionally pivot away from some of our original goals. Other objectives ended up being more complicated than originally anticipated, and we simply did not have the skills or time needed to accomplish them. This is why some of the lower priority objectives were not completed by the end of this project.

Although some of the lower priority objectives were not completed, we did achieve the higher priority objectives, and we delivered a website identical to the design comps exactly as 7Factor asked. We poured time and effort into this project to flatten the learning curve of the various technologies, consistently deliver impressive work to 7Factor, and achieve our milestones in a timely manner. In about 21 weeks, we were able to build a fully developed React website, which had many impressive features given the time frame of this project.

We met about once to twice a month with 7Factor to give them progress updates and to ask them clarifying questions regarding our development. Our communication with 7Factor was a vital factor in the completion of the project and their feedback was always highly appreciated. They were constantly impressed by our progress, and when they did have some suggestions they voiced it to us so we can make the edits accordingly. In total, the feedback and communication from 7Factor was positive.

The website pages, content, styles, and behavior must exactly match that specified in the design comps, unless otherwise specified by 7Factor.

We started out with an original design provided by 7Factor at the beginning of this MQP project, which are attached as separate PDFs to this paper. We were able to fully implement the designs of most of the pages by early C term. However, 7Factor came to us in the middle of C term with updated design changes that they created. Most of these changes were basic, in that there was just text that needed to be added, a button or two added, etc. There were also some more complicated changes that would need a lot more time than we were given, such as adding new sections of pages, adding image vectors, more animations, etc. Due to time constraints, we were unable to complete all of the updated changes that 7Factor gave us, but we were able to complete the higher priority ones.

The website need not be completely responsive, but it should look reasonable on different screen sizes and mobile devices.
We achieved this objective by scaling all the elements in a page by viewport width (VW), aka the width of a user’s browser. This allowed us to shrink and grow a page and its elements responsively, and have the site look good on both mobile screens and extra-large monitors. All the content scales proportionally, and we made sure to put size limits on responsive text so that nothing became so small as to be unreadable or so large that the site appeared unprofessional. This method of exclusively using VW to scale sites is not completely typical: usually sites would have dedicated designs for the mobile version of sites to take different interactivity, screen orientation, and overall size into account. However, 7Factor did not have any mobile designs and expressed that it should simply be presentable on a mobile device, so we chose this route.

The website must be reasonably backwards and cross compatible across browsers, as well as be acceptably efficient when running / loading.

To ensure the website would run efficiently, we followed web design best practices, which included resizing the extremely large assets we were given as to not affect the performance of the website. To ensure compatibility, we tested the website on Firefox, Safari, Chrome, and Opera. In doing so, we were able to identify and fix minor bugs so that the site looks clean, professional, and identical to all users. After these tests we feel confident that our site runs smoothly and looks good on most browsers and devices.

The website should be developed using 7Factor’s Continuous Integration/Continuous Deployment (CI/CD) and workflow management systems.

We were able to develop the website using 7Factor’s Continuous Integration/Continuous Deployment system, where a unit test was run on the website each time code was pushed to the “main” GitHub branch to ensure it was working properly. Once that test passed successfully, the CI/CD system would automatically deploy the changes to the site’s AWS S3 bucket. Using these tools allowed new code to be constantly pushed into production, while preventing any broken code from negatively affecting it. Learning how to use 7Factor’s workflow management systems was a learning curve, but with their assistance we were able to successfully develop the site in this way, and gain valuable experience working in a DevOps environment.

Our design and technical decisions, development and testing processes, workflow, milestones, and conventions must all be documented and source-controlled, as well as shared with 7Factor.

We did most of our inter-team documentation within Google Documents, which included our schedules, technical decisions, milestones, meeting minutes, and more. We had bimonthly meetings with 7Factor to show them our progress, discuss our technical decisions and receive feedback. For other items, we also used GitHub, Clubhouse, and Slack to properly document our
progress and communicate with each other. GitHub is a version-control tool we used constantly to check up on and share each other’s code, as well as make changes and approvals when necessary. 7Factor was able to see our development through the GitHub page and approve tasks such as pull requests. They sometimes made comments to us about our code in these Pull Requests, which allowed for more communication. Slack was a communications platform that we used frequently to message each other, Professor Cuneo, and the 7Factor team. It helped us coordinate meetings as well as receive quick answers to questions we had. Clubhouse was used to delegate tasks for each other and keep up deadlines.

The technical implementation of the website should be adaptable enough to allow 7Factor’s developers to extend the website’s functionality (i.e. dynamic content from a dedicated backend) beyond the requirements of the design comps for this project.

When we first spoke with 7Factor regarding the website, they specified that they had hoped for a backend for the website. They wanted a dynamic content management system, where they could seamlessly add and remove content from a backend. We listed it as one of our objectives for this project. However, within a month or two of the development of this project, 7Factor said that it was not a high priority item, and they would rather worry about the backend themselves. With this new information, and our time constraints, we did not end up developing a dynamic content backend for the website.
Future Work

Case Studies

7Factor wants to continue to advance the development of their Case Studies page, with an in-depth page for each case study. On these in-depth pages, some key content will be the challenges they had to overcome, the solutions they came up with, and the outcomes of those projects. Doing so will allow for potential clients to have a better understanding of how 7Factor tackles problems and the wide range of companies they have worked with.

7Factor also wants to build a CMS backend for this page. Building a CMS backend will allow for 7Factor to quickly and easily add new content and change the current content, always ensuring the case studies they display are up to date and relevant without having to have extensive knowledge of the backend. The CMS backend can additionally benefit the entire website if 7Factor wanted to implement it everywhere. Implementing it site-wide will allow for much easier maintenance and accessibility for people in the company to be constantly updating and adding to the website.

New Pages

7Factor wants to have a ‘Meet the Employees’ page on their website, as one of their core values is that their teams are full of real, human people, not just names. This page would allow for potential clients to learn about each member of the team and allow 7Factor to emphasize the human aspect of their business. Each employee would have their own triangle on the page with a picture of them, and mouse over the triangle would reveal their avatar, name, title, and a short little blurb. Clicking on the triangle will bring up a pop up with their full bio. At the bottom of the page will also be a list of their locations with pictures of each office and a little blurb about each location.

7Factor is also planning on creating a ‘News’ page on their website to allow potential clients to see recent events involving 7Factor, allowing clients to feel more connected to the company by giving them a peek into their operations and lives. Each news article will be given a label, and at the top of the page there will be a label tag so the user can sort by articles with that specific label. There would also be a search feature so clients can further narrow the displayed news. The news articles will all be displayed initially in a grid layout with a picture related to the article and a little snippet of the article, and users can click a ‘Read More’ button to continue to the full news article. On each article there will be a comment section at the bottom to allow people to leave a comment, their name, and their email, allowing intractability between the company and the client.
Mobile Design

While we have created fully responsive pages on browsers using grid systems and scalable elements, we have not been designing the website with mobile phones in mind. Having a dedicated design that takes the unique interactivity and layout of mobile devices into account would be very useful for increasing the accessibility and presentation of the website.

Accessibility support

As of right now there is only a minimum amount of support on this site for accessibility for people who are impaired in any capacity. Adding in various support capabilities, such as a color-blind mode, support for users with a larger default font size, or text-to-speech would allow for even greater ease of access to everyone and would show how much 7Factor cares about their clients.

Full Migration to Hooks

We recommend that 7Factor fully migrate the website to React Hooks. Migrating to hooks will maximize the utility that comes with using React.JS and will allow for much cleaner code, for it will be more organized and easier to understand and use. Properly integrated hooks would break down the website into little code compartments, and in doing so, these compartments can be easily duplicated throughout the website if necessary, such as the floating menu and the header. Additionally, it would allow multiple developers to work on different aspects of the website simultaneously without having to worry about merge conflicts.

Formal Testing

While 7Factor did recommend that we use unit and integration testing, due to time constraints and the low priority 7Factor placed on it, we didn’t end up implementing either. However, we do recommend that unit testing be implemented to properly test out the code in order to make sure the code doesn’t break. Properly implementing unit testing will ensure that any future code added doesn’t break any existing code. We also recommend integration testing be integrated for the similar purpose of making sure the website overall still works after new code is added. Both types of testing should continuously be updated with new tests to ensure all aspects of the website will always work when new code is added.

Better Ticketing and Rigid Workflow Methodology

While clubhouse and agile methodologies are incredibly useful for keeping track of and breaking down tasks, as well as setting clear weekly goals, we felt that as a team we were small enough and communicated well enough to forego a ticketing system and a rigid agile approach to this project.
However, if a larger team were to work on this project we do recommend using clubhouse and agile methodologies because they are incredibly good at structuring team tasks and keeping them on track to finish work in a timely fashion.
Bibliography

7Factor, LLC. “7Factor/7factor.” GitHub, 26 Feb. 2019, github.com/7Factor/7factor.


Glossary

**Agile** - Agile is a software development practice that is used to create a guide for engineers to program applications. Agile can overall be described as iterative requirements gathering and development process, and includes practices such as daily “stand-up” meetings and “sprints” of high-speed development for a specified amount of time.

**Application Programming Interface (API)** - a software interface that mediates and enables communication between two different software applications.

**AWS S3** - A cloud-based storage bucket hosted in Amazon Web Services which allows for easier storage.

**Backend** - The part of the software hidden from the client that typically handles data access from APIs and databases, or hosting the frontend.

**Backwards-compatibility** - The ability for an application to be seamlessly integrated with older systems.

**Continuous Integration/Continuous Deployment (CI/CD)** - The practice of frequently committing code to version control and using automation to test, build and deploy code changes.

**Content Management System (CMS)** - a backend functionality or software that allows content to easily be created, modified, and published on a website. Typically used in blogs or news feed pages.

**Codebase** - The primary source of code for a project or company, usually stored in a version control system such as a GitHub organization or repository.

**Cross-compatibility** - The ability for an application to look and perform identically on different systems (in this case primarily web browsers).

**Call to Action (CTA)** - Any element, but usually a button in a web application, that encourages a user to take a positive action - explore the site further, fill out a contact form, etc.

**Design Comps** - a high fidelity mockup or wireframe of a proposed design for a web application.

**DevOps** - A practice closely relating to Agile methodology that leverages tools and philosophy to produce high quality software while accelerating the software development lifecycle. DevOps
usually include automated pipelines and toolchains that allow software to be quickly developed, tested, and deployed to production.

**Flask** - A Python-based microframework for web development. It is categorized as lightweight, flexible, and scalable without many built-in features.

**Framework** - A language-specific skeleton that dictates the structure and functions of an application. A platform for developing software applications.

**Frontend** - The part of the software that users see and interact with. In this project the frontend primarily consists of code built using the React framework.

**HTML Forms** - A form used to collect user input from a website, where it is then sent to a server, email, or online data storage system.

**Hypertext Markup Language (HTML)** - A markup language used to display documents on the internet. React is loosely based around HTML tags.

**JavaScript** - A programming language primarily used in web development that conforms to the ECMAScript programming language.

**Package** - A collection of third-party code that can be integrated into a software project from which various functions and features could be used.

**React Router** - A tool in React.js that allows you to handle routing in web apps, using dynamic routing. Dynamic routing takes place as the web app is rendering on your screen.

**React Switch** - A React library which declares the routing of web applications. This library renders routing exclusively.

**Representational State Transfer (REST) API** - The architectural style of API that adheres to best practice principles such as separating client and server functionalities, maintaining a stateless environment, and maintaining a uniform interface.

**SCSS** - a superset of CSS that allows for nested rules, variables, and more; a more advanced version preprocessor language that easily compiles CSS.

**Stakeholders** - people who have a vested interest in the success of the software project such as clients or investors.
**SVG** - A vector image format with support for animation and interactivity.

**Target audience evaluation** - A UI/UX practice in which the target audience of a system is identified and select members evaluate to product for design, usability, or accuracy.

**Tech Stacks** - A list of all the technology services used to build and run a single application.

**Trunk based development** - A practice used in version control in which work is branched directly off of the main branch and features are merged in as quickly as possible to reduce merge conflicts and ensure fast development.

**Search Engine Optimization (SEO)** - The optimization of a website to improve the quantity of web traffic from popular search engines.

**Software as a Service (SaaS)** - A software delivery model where customers purchase a software package that is hosted and maintained by a third party.

**User Interface / User Experience (UI/UX)** - The practice of user-driven design that places the experience of the user as the top priority.