Creating the New Face of Buildertrend

Design Document

Team: sddec20-07

Client: Buildertrend

Advisor: Mai Zheng

Mitchell Knoth - Team Meeting Scribe

Melissa Hernandez - Team Communications Leader

Dallas Evers - Team React Expert/Code Reviewer

Joshua Bump - Team Webmaster

Kailyn Mann-Calaway - Team Reports Manager

Darnesha Randle - Team Leader

Email: sddec20-07@iastate.edu

http://sddec20-07.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

- Write clean, well documented, modular code
- Follow the Buildertrend React migration progress
    - Work with BT dev, product, and QA teams
- Follow Buildertrend frontend design guidelines
- The React pages should be composed of high quality, correct, bug-free code

Summary of Requirements

- The team must adhere to all confidentiality agreements set by our client
- Meet with the client once a week
- Maintain communication with Buildertrend teams
    - Microsoft Teams
    - Pull Request Comments / Suggestions
- Each team member must contribute at least 3 hours of work a week
- Tech Discoveries must be done before each page is created
- The team must test original pages and log the results
- Build React common components
- Use the given Buildertrend API

Applicable Courses from Iowa State University Curriculum

- Com S 227: Object-Oriented Programming
    - This course provides students with an introduction to object-oriented programming, including the four pillars of OOP: Encapsulation, abstraction, inheritance, and polymorphism
- Com S 228: Intro to Data Structures
    - This course introduces the importance of data structures in computer programming, covering some of the most widely known and used data structures including linked lists, graphs, trees, and stacks. 228 also focuses on big-Oh notation and sorting algorithms.
- Com S 309: Software Development Practices

    - This course helps students to develop teamwork skills on software projects by giving them a semester-long project to work on. Through this course, students are introduced to the Agile Approach to Software Development and developing an application that can be useful to others.
- S E 319: Construction of User Interfaces
    - This course introduces students to different languages and frameworks that can be used for the development of user interfaces. In this course, students learn things such as HTML, JavaScript, Node.js and similar technologies. This course also allows students to learn about the Software Development process through coursework and a small group project in the last month of the course.

New Skills/Knowledge acquired that was not taught in courses

- React Development
- Conversion of legacy code to new platform
- Testing, quality assurance, and documentation of previously written code (Tech Discoveries)
- Communication with client

Table of Contents

List of figures/tables/symbols/definitions

Figures

Tables

1. Introduction

1.1 Acknowledgement

This project was sponsored by Buildertrend Solutions. Our team received technical assistance and guidance from Buildertrend's Senior Architect Rick Kalasky, Project Manager Alli Kellner, and Software Developers Daric Teske and Cameron Hessler.

1.2 Problem and Project Statement

General problem statement: Buildertrend is currently in the process of trying to improve the speed and stability of their application website. To do this, Buildertrend is converting their outdated code into something newer, faster, and more efficient through the help of ReactJs. This conversion is taking time though, which is straining Buildertrend's team, seeing as they must currently support two different front-end systems.

General solution approach: Our senior design team will be responsible for building as many of Buildertrend's new website pages as possible. Our team will be given access to the existing Buildertrend pages and will be responsible for reverse engineering a minimum of 3 pages into ReactJs. The pages our team creates for Buildertrend are expected to replace the older pages and will go live on the company's website by the end of the senior design year. Our project will greatly help Buildertrend by speeding up the React Migration the company has been undergoing for about a year now.

1.3 Operational Environment

Buildertrend provides its clients with software meant to simplify and optimize the building experience for homeowners, contractors, and remodelers. The environment in which our pages will be used will primarily be on an online application, accessible through laptops, desktops, and mobile devices. Due to the nature of most technology applications, our pages must be quick, reliable, stable, and thoroughly tested. We wish to avoid pages that crash or lag often, seeing as we want the users to have a pleasant experience using the Buildertrend application.

1.4 Requirements

1. A Tech Discovery shall be completed for each web page we recreate to capture all of the requirements of the page we are recreating.
2. The recreated web pages must function exactly as the original web page.
3. The project shall be completed utilizing React Components when dealing with the User Interface recreations.
4. We will be required to keep our code on an internal Buildertrend Git repository
5. We will be required to use Visual Studio and Visual Studio Code as we develop our code.
6. Testing must be done utilizing Storybook and React testing libraries.
7. Some User Interface components must be designed using the Antd library.
8. We will be required to have bi-weekly meetings with the Buildertrend team.
9. We must recreate at least three web pages for Buildertrend.

Note: no other requirements exist.

1.4.1 Engineering Constraints and Non-functional Requirements

Constraints

1. The pages shall be completed by the end of Senior Design in November 2020.
2. We will be required to implement the solutions utilizing APIs that Buildertrend already has in place at their company.

Non-functional Requirements

1. The recreated web pages must look exactly identical to the original web page.

1.5 Intended Users and Uses

Intended Users	Intended Uses
Homebuilders	1. Everything you need to manage the home building process from start to finish. 2. Turbo charge your efficiency with the Buildertrend platform. 3. Customer management tools that allow you to keep your customers up to date on the project. 4. Project Management tools that allow you to easily view documents, mark up plans, manage scheduling, send estimates and more. 5. Financial tools that allow you to set budgets, send purchase orders, formalize pricing and manage the lifecycle for your projects.
Remodlers	1. Renovate your business model with a full suite of tools and resources. 2. Connect with your crew, keep your customers happy and grow your business with help from Buildertrend. 3. Customer management tools that allow you to keep your customers up to date on

	the project. 4. Project Management tools that allow you to easily view documents, mark up plans, manage scheduling, send estimates and more. 5. Financial tools that allow you to set budgets, send purchase orders, formalize pricing and manage the lifecycle for your projects.
Speciality Contractors	1. Solutions that are adaptable, intuitive, and focused on boosting profitability. 2. A single, powerful platform is all you need to expertly manage all aspects of your business. 3. Customer management tools that allow you to keep your customers up to date on the project. 4. Financial tools that allow you to set budgets, send purchase orders, formalize pricing and manage the lifecycle for your projects. 5. Pre-sales process tools that allow you to manage new leads, win more bids and streamline your sales process with tools that belong on a superhero’s utility belt.
Commercial Contractors	1. Grow your business with powerful project management tools. 2. Take your business to the next level with software that is as ambitious as you are. 3. Project Management tools that allow you to easily view documents, mark up plans, manage scheduling, send estimates and more. 4. Financial tools that allow you to set budgets, send purchase orders, formalize pricing and manage the lifecycle for your projects. 5. Pre-sales process tools that allow you to manage new leads, win more bids and streamline your sales process with tools

	that belong on a superhero's utility belt.

Table 1. Intended users and uses

1.6 Assumptions and Limitations

Assumptions:

- Design (color scheme):
  - All web pages will follow the required Buildertrend company color scheme.
- Methodology:
  - The project will be completed using the tools and techniques prescribed by the Buildertrend methodology.
- Technology:
  - The project will use the following technology, tools, and libraries: React, Formik, Storybook, Ant Design, and TypeScript.
- Product environment:
  - All web pages must be accessible through the Buildertrend website and able to be viewed on laptops, desktops, tablets, and mobile devices.
- Product performance:
  - The maximum number of simultaneous users / customers is unknown at this time.
  - The number of users / customers varies greatly depending on the season.
  - The maximum number of simultaneous users is assumed to be 100.
  - This is important because we want to ensure that our pages are quick, reliable, and stable.
  - If we underestimate the number of simultaneous users then the pages may crash or lag often.
- End product geographical use:
  - The end product will be used all over the world
- Team performance & skill:
  - Each team member is expected to have the necessary technological skills to be able to contribute to the project in a meaningful way.

Limitations:

- Budget & finance:
  - The cost of completing the product shall be no more than $500, which is the allotment given to us by departement for senior design projects.
- Schedule:

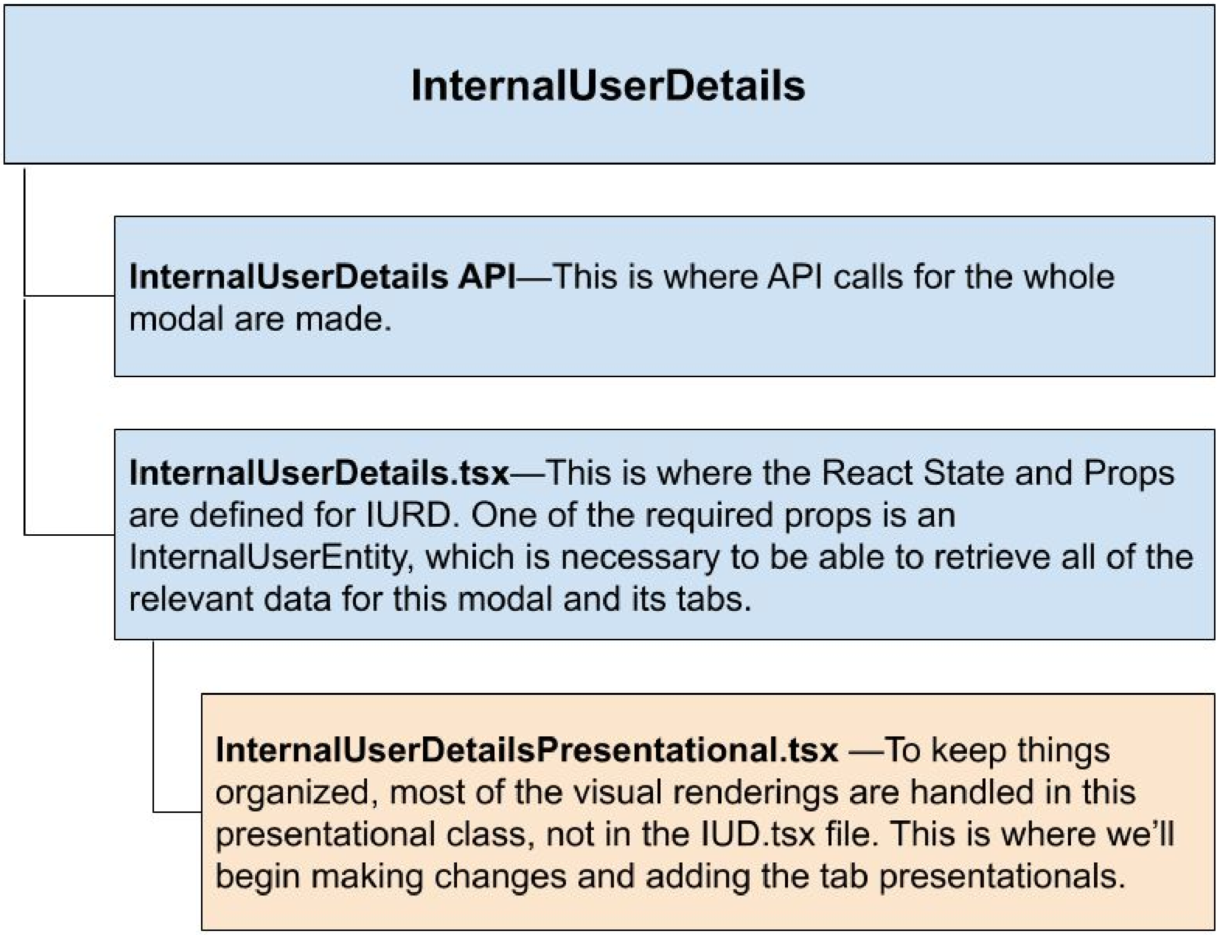
  - The project must be completed by the end of the senior design year in November 2020.
- Scope:
  - Responsible for reverse engineering a minimum of 3 pages into React.
    - Each page requires a Tech Discovery.

1.7 Expected End Product and Deliverables

At the end of our senior design product, we will have recreated at least three web pages for Buildertrend. Each web page should look and function as close to the original web page as possible. Each page is assigned to us by Buildertrend and we were given what Buildertrend deems their expectations for us with regards to the completion of a successfully migrated React page. All documentation regarding the web pages will be commented within the code. The web pages will be delivered periodically throughout the Spring and Fall 2020 semesters; Our first page that we were assigned to fully convert was the Data Science page, and in the fall we were given our last pages to convert: Job Access Page, Permissions Page, and Preferences Page. All these pages were tabs under the Internal User Detail Modal.

2. Specifications and Analysis

The overall design on the react pages that we will be converting all follow a similar structure. For example our Internal User Details model has the following architecture:



InternalUserDetails

InternalUserDetails API—This is where API calls for the whole modal are made.

InternalUserDetails.tsx—This is where the React State and Props are defined for IURD. One of the required props is an InternalUserEntity, which is necessary to be able to retrieve all of the relevant data for this modal and its tabs.

InternalUserDetailsPresentational.tsx —To keep things organized, most of the visual renderings are handled in this presentational class, not in the IUD.tsx file. This is where we'll begin making changes and adding the tab presentationals.

Figure 1. Architecture diagram of Internal User Details.

The API portion of the architecture is where all the API calls are made that are related to that specific page. It is how we are getting the data we need to populate the React Component with. Going into our InternalUserDetails.tsx file, this is where all the React components come together. We created the layout of the page using React components in this file and passed along the required and relevant data through the props and state in order to populate some of the UI components. The presentational layer is used for rendering and keeping things organized that are not being done in the InternalUserDetails.tsx file.

Because we were working with a client, there was actually very little additional planning done by our team. Our instructions were to rebuild existing .NET pages using React, while keeping all of the functionality and the layout mostly the same. There were a few exceptions, usually handed to us by the UX team with details as to what they wanted to change, but for the most part, we were following this code structure and doing as we were told.

2.1 Proposed Approach

Our planned general approach to this project was to follow Buildertrend's Tech Discoveries guidelines on existing Buildertrend web pages (more details in section 5.1). Tech Discoveries allow our team to essentially do black-box testing on the existing pages, which allow us to not only familiarize ourselves with the functionality of the page, but also to find any existing problems or bugs that had gone unnoticed. Our senior design team also coordinated with Buildertrend's Code Health team to make sure all of the required APIs for our pages exist and are functioning correctly.

After the Tech Discovery had been completed and the page had been fully documented, our team could begin to reverse engineer the pages using ReactJs and the APIs provided by Buildertrend's team. . This process varied depending on the page we had been assigned. During our spring semester, our team was given a page to build from scratch. This might sound like a difficult process, but as we later found out, it was actually easier to start from scratch. Our team was able to design the page, the API types, and the React components as we saw fit. It gave us a lot of flexibility and room for customization.

The pages we received in the fall semester were quite different. A group of summer interns had already started some of the pages and our job was to finish the remaining pages and integrate them with existing code. This proved to be a tricky and time consuming process. Our team had to read and understand all of the code that directly tied into our pages—the modal that would be passing in data to our pages, for example—before we could begin working. Some common components already existed in Buildertrend's code, some did not. We were not free to write the code in any way we wanted like the first semester. It was first necessary to understand the code that was already there so that we could create our code to fit in easily with the rest of the code base. This felt more like the type of development we will have to do in the real world someday, so it was a valuable learning experience for our team.

Additionally, for our first semester working on this project, Buildertrend was using Bitbucket as their source code management system and followed a straightforward process. Once our newly created ReactJs pages had been pushed to the Buildertrend Bitbucket repo, a React Migration manager code reviewed our team's work. Once our pages had been thoroughly tested, documented, and reviewed, our newly migrated pages were deemed “Dev Complete” and passed on to a Code Health team member. Had Code Health found any significant problems with our pages, they were returned to our senior design team to either edit or fix the work. Otherwise, Code Health essentially finished the page and we were able to start on our next migration. For the second semester our process was very similar, but differed because we were using a different source code management system (TFS) because Buildertrend had made the switch over the summer.

2.2 Design Analysis

During our first semester our team had been road blocked for most of the semester, first by non-disclosure agreements and then by our assigned pages being taken by another team, we finally were able to start on our project. During our meeting with Buildertrend the week before spring break, Daric informed us that we have our first page: a Data Science Information (DSI) page. Our team members have been working to set up our machines by cloning the Buildertrend Bitbucket repository, installing our Amazon Web Services VM, and familiarizing ourselves with the Buildertrend app. We had begun our Tech Discovery for the DSI page at the time, and we continued over the course of the next following couple of days. Once the Tech Discovery was finished, we created a branch in the repository and began assigning tasks on our Trello board.

As we moved forward, communication and organization was going to be key for our project, especially considering we worked remotely indefinitely going forward. It will be important for our team to check in often and frequently, so we can avoid stepping on each others’ toes or overwriting each others’ code. Our proposed plan of working in one page branch and creating smaller branches off of this one in our repository seems like a strong plan as of now. Daric approved of this idea and it seems this will be the best way to keep our work clean and organized.

Our architecture of our project will be coherent with Buildertrend’s architecture that they have in place for all their applications. The React pages that we work on will be created on a premade API that is made for us by Buildertrend. We will stick to what development practices Buildertrend outlines us to follow. Our expected results will be, at a minimum, three pages fully converted by the end.

2.3 Development Process

Our team has been using an agile approach. We have treated our Trello board as our scrum board, which was being updated regularly with tasks and cards that our team needed to complete. Each task was assigned to a team member, and everyone will be responsible for marking their tasks complete once they are finished.

Our team also used Bitbucket/TFS to organize our work. Each of our team members had cloned the repository Buildertrend created specifically for this React Migration. From there, we created a branch for each page we worked on, and we had branched off of our page branch for any individual tasks we needed to complete. This method allowed us to avoid merging incomplete work into master, which would have potentially produced negative side effects for the rest of the repo. Once a page is fully complete we can submit a pull request, which will need to be approved by Daric. If our work is clean, bug-free, and meets Buildertrend's coding standards, our page branch will be merged into master and our team will be able to move on to the next page.

Our team went with an agile approach due to the flexibility and adaptability this type of development process usually allows. Every page we were assigned were all different, with some much larger and requiring much more work than others. Because of this, our team felt it would be best to do iterative development. After each iteration, our team would have new working code and more progress made than if we tried to do waterfall or TDD.

2.4 Conceptual Sketch

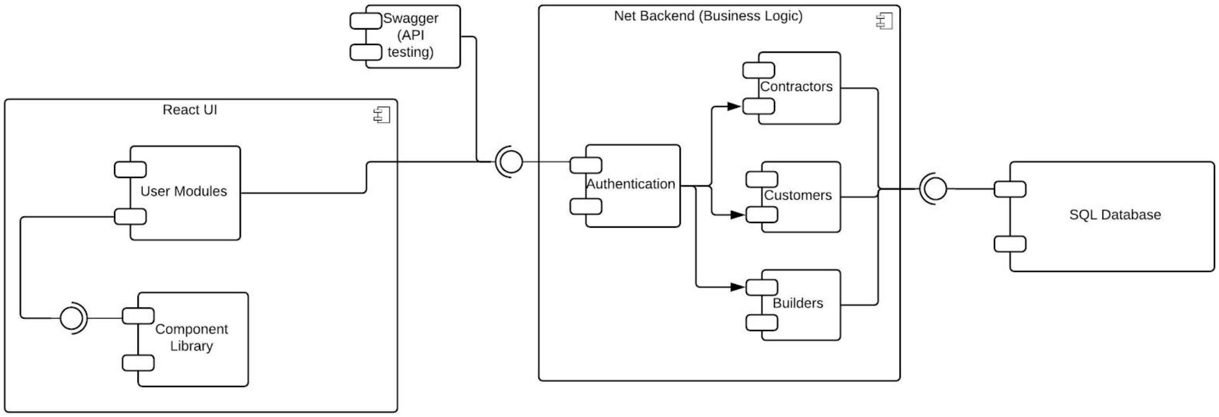


Figure 2. Conceptual sketch of our project

3. Statement of Work

3.1 Previous Work And Literature

The existing work that we had been following had come directly from the Buildertrend application itself. Since we are not actually creating anything new, our work will not be affected or influenced by other products in the market. We have been given existing Buildertrend web pages that we must recreate using ReactJs. The advantage to this is that our team does not have to worry about designing any new pages or building APIs for these pages to use. The layout, design, and API of each page we built had already existed. This puts less of a burden on our group and reduces the chance of failure in our project. Due to our limited work experience, creating and designing a page from scratch could have been difficult for our group or would have created more obstacles.

The disadvantage of this project is that the design experience we could have gotten had we been assigned to build a page from scratch would have been valuable to each of our team members. In addition, our creativity when working on these pages will be slightly stifled, considering the pages we are building already exist, simply in another coding language. We were not allowed to change much about the pages. We were only supposed to rewrite them using ReactJs.

3.2 Technology Considerations

The biggest advantage to using ReactJs is its speed. React is known for using "components," which are small fragments of a web page, such as a modal or a nav bar. React will update only those components which have been modified instead of reloading the entire page, which makes React a very quick and efficient language to use for websites. This was the entire motivation behind using ReactJs to replace the current Buildertrend web page code.

3.3 Task Decomposition

- Get web page
  - Complete Tech Discovery where we analyze the page to see its setup and how it performs.
  - Split up the web page into equal sections for each team member to complete on their own.
    - Each team member will have a branch from our page's branch in Buildertend's repository.
    - We will each be responsible for merging our branch to the page's main repository once our work is complete.
- Completed web page
  - Once all the tasks are complete for the page branch, our team will submit a pull request to Daric.
  - If there are no problems with our work then Daric will merge the branch into master.

3.4 Possible Risks And Risk Management

Some possible risks for this project involve knowledge of the required coding languages. Failing to know any of the required languages or how to implement something in any of the required languages will hinder our progress. Additionally, if pieces of the code are implemented in inaccurate or nonoptimal ways, this will also slow our progress. To handle these risks, we continued to learn the required languages while we were waiting to begin coding. If we ran into any of these issues, we reached out to other teammates for help, or asked for assistance from our client.

3.5 Project Proposed Milestones and Evaluation Criteria

Each milestone is the completion of a page. The pages are tested while we are coding and after it has been completed; we run the web page to ensure it looks and is operating in the exact ways the original web page looked and operated. Our team and Buildertrend had decided that it was not in our project scope or responsibility to be writing any tests for the web pages.

3.6 Project Tracking Procedures

Our team had tracked our coding progress through a Trello board and tracked how much time each individual had put into the project through our bi-weekly reports. Additionally, we had given each other and our client updates periodically through our group chat and during our bi-weekly meetings.

3.7 Expected Results and Validation

By November 2020, we were expected to have recreated three to four web pages for Buildertrend. These web pages should look and behave exactly like the original web pages Buildertrend gives us. We confirmed our results are high level by checking that the user interface and performance of our recreated web pages perform exactly as the original web pages do. We used our Tech Discoveries to do this, as well as Storybook to validate that our code worked and met expectations.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

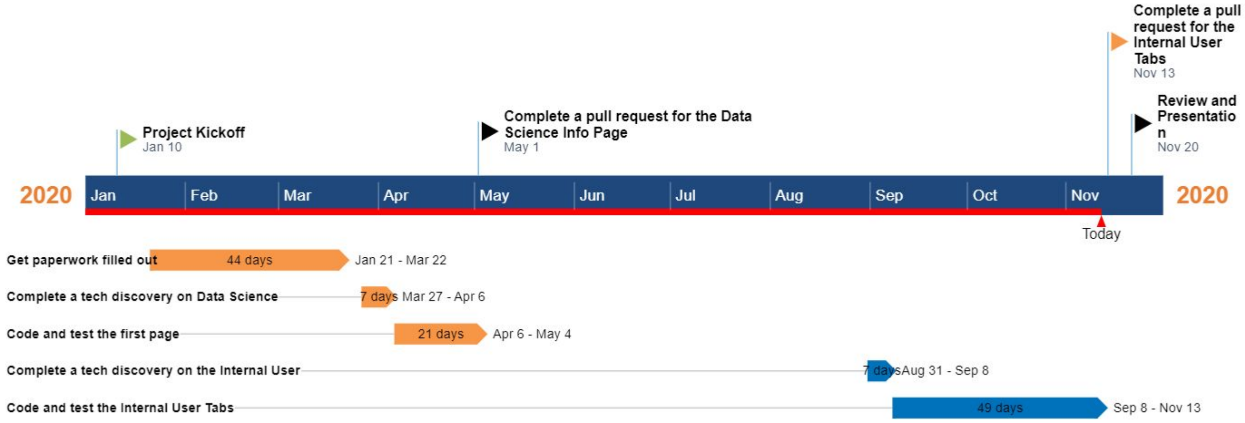


Figure 3. Project timeline

This timeline had been changed as time went on and how Buildertrend assigned us a modal with many nested tabs. Each tab according to Buildertrend will suffice as a web page and Buildertrend expects us to finish at least three web pages, converting them from ASP to React. The Gantt chart above shows the dates the three pages should tentatively be completed. A breakdown of why we chose these dates is below. As this project comes to a close we were able to keep up with our Gantt chart and complete the three pages in a timely fashion. As shown in the chart, it took quite a long time for us to receive access to the systems from Buildertrend, and we are starting the web page development a little bit later than we would have liked.

Each page had three steps to complete, ending with a pull request on Git. For each page, we had documented the page we were recreating, accounting for all of its features, in a Tech Discovery. This process took around a week to complete for our first page, but for our last batch of pages the Tech Discovery was mostly completed by the summer interns. After we had a finished Tech Discovery, we began work on the page as a team using multiple branches on Git. We were testing as we went. This took around three to five weeks to complete. We have completed all of our tasks and we are still in the process of making a final merge and completing a pull request into the pipeline for Buildertrend.

4.2 Feasibility Assessment

With our progression through the first semester, our team was fully integrated into Buildertrend's network and workspace. However, when we made the transition into our final semester we were faced with countless virtual machine problems, which were the result of Buildertrend switching over their SCM to a new provider. Our Virtual Machines were not successfully migrated over accordingly. Additionally, for most of us who have not worked at Buildertrend before, it took some time to get adjusted to the development process that they have in place. This ranges from testing, working with new platforms, and documenting Tech Discoveries. Once our team became adjusted to this new environment, we were able to deliver at least three fully migrated React pages.

4.3 Personnel Effort Requirements

Task	Time (hours to completion)	Projected Effort (scale of 1 (little effort to) to 5 (most effort)
Research (learning how to use React, Formik, Storybook, Ant Design, and TypeScript)	30 - 50 hours	Avg 3 (depending on the member's familiarity with the tools it may take more or less effort)
Tech Discoveries (need to be very detailed documentation about the interactions, API calls for each page)	10 - 20 hours total per page	Avg 4 (because these task requires a lot of detail it will require more effort)
Development (Programming pages, Git Commits)	20 - 60 hours total per page (depending on how large the page is)	Avg 4 (the team works simultaneously on each page so having each page being worked on by 6 people will be hard)
Testing (testing to see if each page performs the required actions)	3 - 7 hours total per page	Avg 2 (use black box testing)

Table 2. Personnel effort requirements

4.4 Other Resource Requirements

Aside from a laptop, there are not any other resources that are required to conduct the project. The project can be completed in its entirety from each group members' laptop using AWS Workspaces.

4.5 Financial Requirements

There are no required financial resources for this project.

5. Testing and Implementation

5.1 Interface Specifications

We are using our Tech Discovery and Buildertrend's original web page to compare our user interface to the one Buildertrend gave us. The process of implementing each page consists of similar steps. First, we complete the Tech Discovery. The Tech Discovery is a process Buildertrend gives us for identifying all the important aspects of an existing .NET page: API calls, common components, functionality. During this process, our team documents the current behavior of the page, as well as any bugs or issues. This step is crucial for allowing future teams to pick up where we left off if necessary or so that Buildertrend may reference our documentation to better understand our work. Tech Discoveries are also extremely beneficial in that they allow our team to learn about the page so that we may better understand our responsibilities and requirements.

After the tech discovery has been completed and the page has been fully documented, our team can begin the implementation. This process varies depending on the page we have been assigned. During our spring semester, our team was given a page to build from scratch. This might sound like a difficult process, but as we later found out, it was actually easier to start from scratch. Our team was able to design the page, the API types, and the React components as we saw fit. It gave us a lot of flexibility and room for customization. The pages we received in the fall semester were quite different. A group of summer interns had already started some of the pages and our job was to finish the remaining pages and integrate them with existing code. This proved to be a tricky and time consuming process. Our team had to read and understand all of the code that directly tied into our pages—the modal that would be passing in data to our pages, for example—before we could begin working. Some common components already existed in Buildertrend's code, some did not. We were not free to write the code in any way we wanted like the first semester. It was first necessary to understand the code that was already there so that we could create our code to fit in easily with the rest of the code base. This felt more like the type of development we will have to do in the real world someday, so it was a valuable learning experience for our team.

5.2 Hardware and software

Our team utilized only software because everything surrounding our project from our perspective was all virtual.

Google Chrome Developer Tools: This software is utilized when our team is flushing out our Tech Discoveries. It helps us see all the moving parts of the webpage we are trying to convert. It also helps us see what exact API calls

Postman: Our team used this software when we were testing our code to see if our API calls were successful and returning the correct information we were looking for. It is a good tool to abstractly test specific functionality and communication within an application.

5.3 Functional Testing

We test the application as we code. We test the code's functionality (according to our Tech Discovery's specifications) by running the code after each method is complete, using Postman to test our API calls. Additionally, the Buildertrend team is overlooking our project and ensuring that our Tech Discoveries fit their expectations. If we implement anything that does not fit Buildertrend's standards, the React Migration team manager will see this in the Tech Discovery and ask us to change it.

In addition, Storybook is a useful tool that allows us to test and experiment with different component types. Storybook allows components to be developed and tested in isolation, which is incredibly useful when it comes to building up our page. With Storybook, we don't have to worry about deploying our website to be able to see what it will look like. As we work through the components and the front end aspect of our page, all we have to do is hit save and Storybook will automatically re-render our components, allowing us to view the changes we made. We consider this tool a testing tool because it allows us to instantly see the effects of any changes we make to components. Furthermore, we can interact with a component on our page to make sure it functions the way we expected. For example, does this button change color when we click it? Is the value of the drop down stored correctly? Etc.

Outside of this we do not have any formal type of tests such as unit testing, integrity testing, or user study. Most of our testing comes from using Postman to test our API calls. We can determine the anticipated test results for each test case by using the information given to us by Buildertrend. We also test our API results by comparing our results against the API results of the current Buildertrend page. If testing shows that we aren't getting the desired results then we will go back into the code and make the necessary changes and then retest.

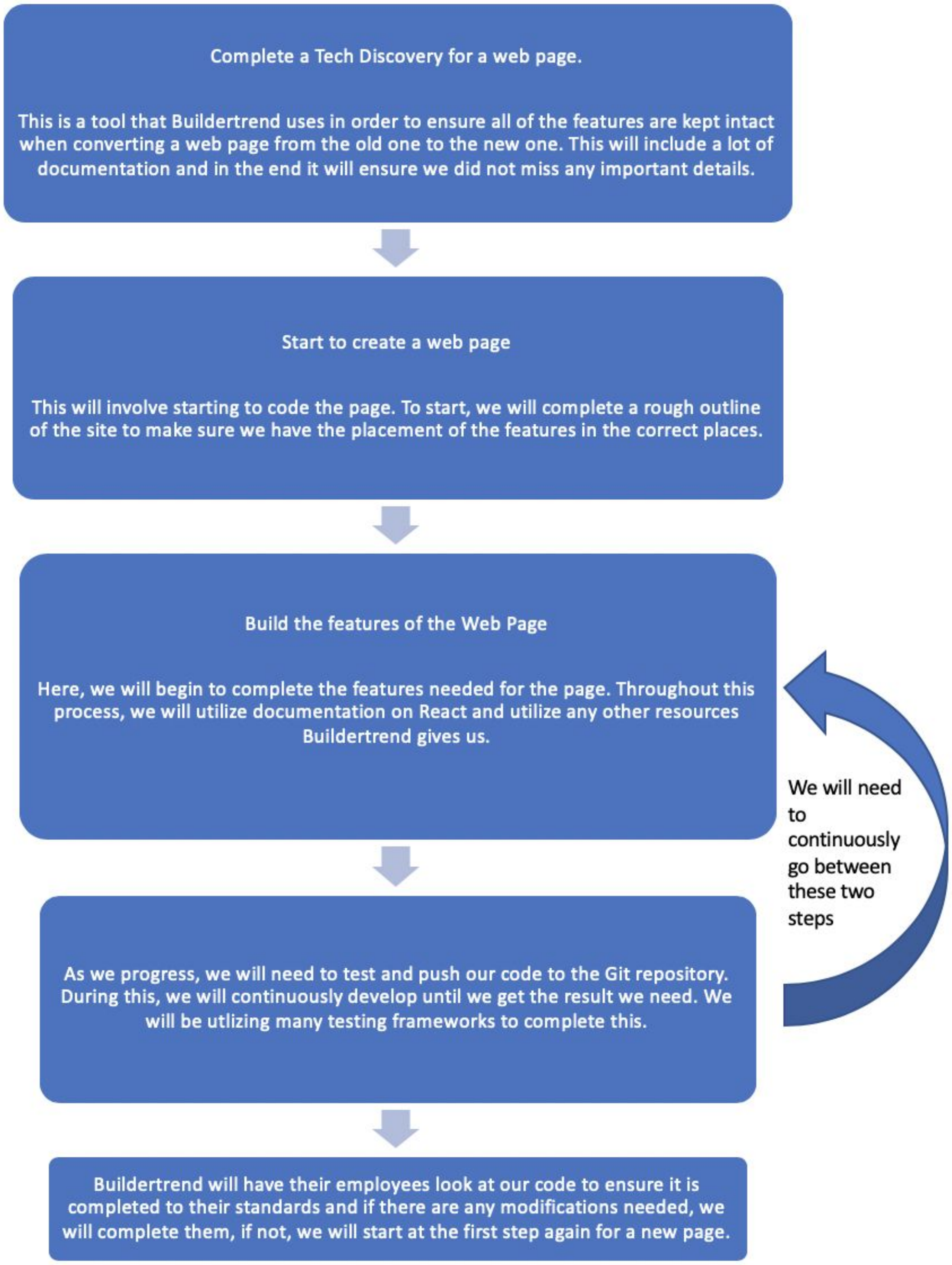
5.4 Non-Functional Testing

We test the performance of our code after we complete a significant portion. We test the performance by making sure that when we perform certain actions, such as selecting an option from a drop down menu that the page is responsive to that change in a timely manner. We also made sure that our code would support different types of users. For the Permissions tab we had to populate dropdowns with all the different types of roles that a user could be assigned to. Based on their level of permission assigned to that user, it would change the view of the overall permissions page and what checkboxes would be set to “true” and what would be “false”.

5.5 Process

The methods indicated in section 2 were provided to us by the client and are something we can neither test nor change. Buildertrend gave us the steps we should take when migrating a page and our job is to follow those steps, which are illustrated below.

Flow diagram of the process:



Complete a Tech Discovery for a web page.

This is a tool that Buildertrend uses in order to ensure all of the features are kept intact when converting a web page from the old one to the new one. This will include a lot of documentation and in the end it will ensure we did not miss any important details.

Start to create a web page

This will involve starting to code the page. To start, we will complete a rough outline of the site to make sure we have the placement of the features in the correct places.

Build the features of the Web Page

Here, we will begin to complete the features needed for the page. Throughout this process, we will utilize documentation on React and utilize any other resources Buildertrend gives us.

As we progress, we will need to test and push our code to the Git repository. During this, we will continuously develop until we get the result we need. We will be utlizing many testing frameworks to complete this.

We will need to continuously go between these two steps

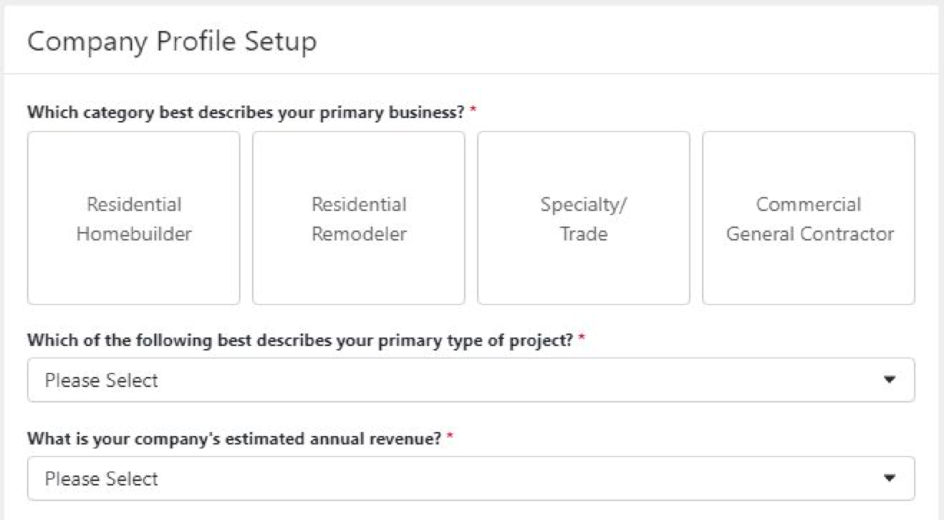
Buildertrend will have their employees look at our code to ensure it is completed to their standards and if there are any modifications needed, we will complete them, if not, we will start at the first step again for a new page.

Figure 4. Flow diagram of the process

5.6 Results

Preliminary Results:

Below are images of each of the current web pages that we have converted to React.



Company Profile Setup

Which category best describes your primary business? *

Residential Homebuilder | Residential Remodeler | Specialty/ Trade | Commercial General Contractor

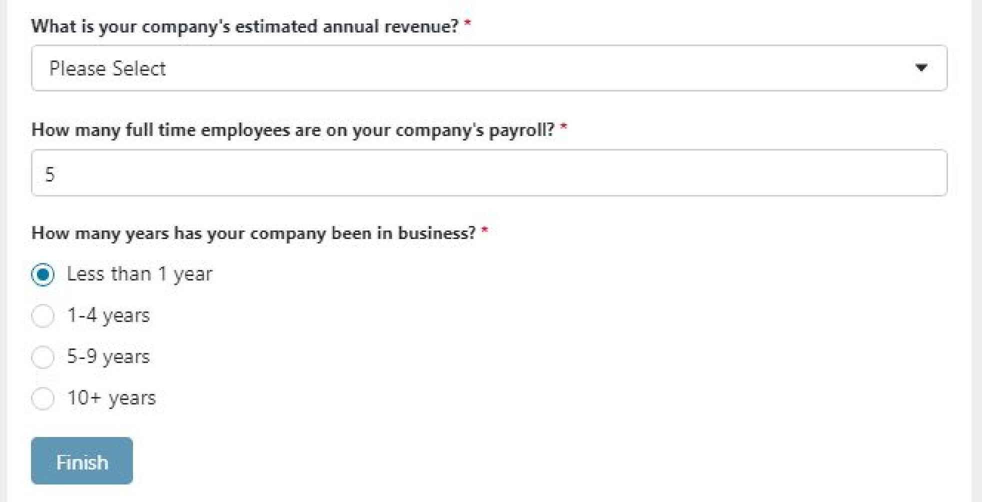
Which of the following best describes your primary type of project? *

Please Select ▾

What is your company's estimated annual revenue? *

Please Select ▾

Figure 5. Current React version of Data Science Information page that was completed in May of 2020 and is now out to production.



What is your company's estimated annual revenue? *

Please Select ▾

How many full time employees are on your company's payroll? *

5

How many years has your company been in business? *

● Less than 1 year

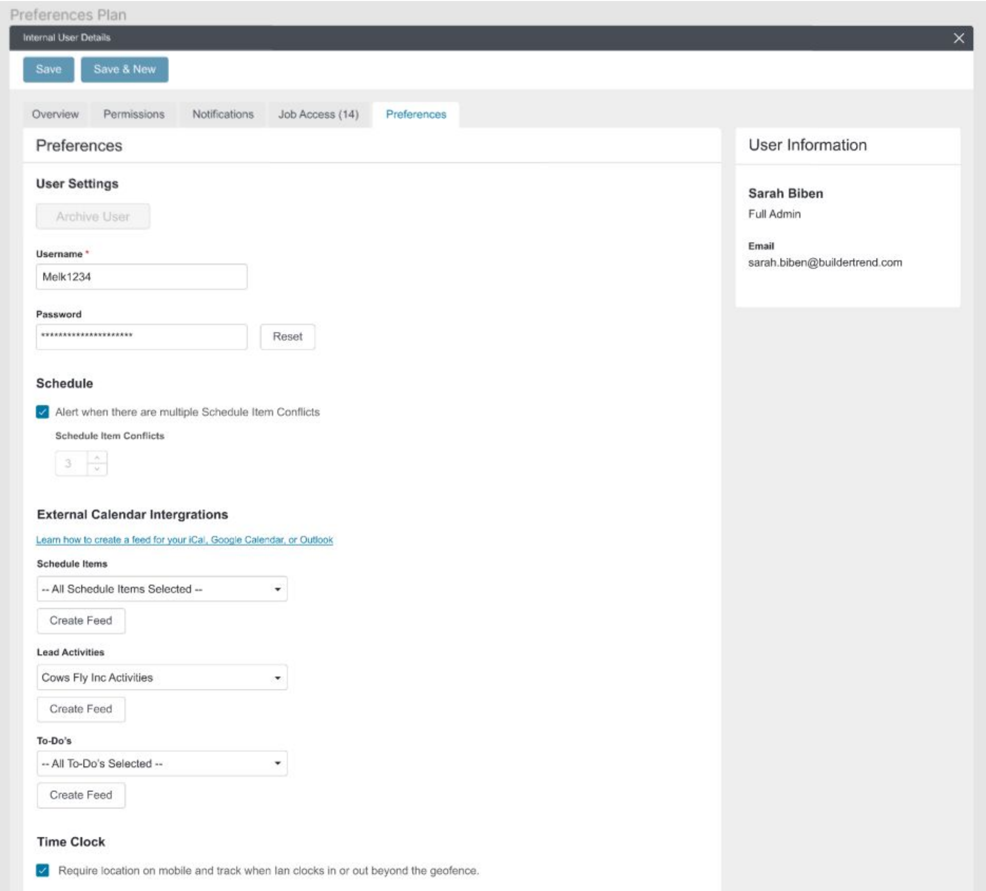
○ 1-4 years

○ 5-9 years

○ 10+ years

Figure 6. Current React version of Data Science Information page that was completed in May of 2020 and is now out to production.

Our team also has successfully converted several of the Internal User Details tabs and tested them utilizing the tools we were given. Below are images of the pages our team has finished.



Preferences

User Settings

Username *
Melk1234

Password
****

Schedule

☑ Alert when there are multiple Schedule Item Conflicts

Schedule Item Conflicts
3

External Calendar Intergrations

Learn how to create a feed for your iCal, Google Calendar, or Outlook

Schedule Items
-- All Schedule Items Selected --

Lead Activities
Cows Fly Inc Activities

To-Do's
-- All To-Do's Selected --

Time Clock

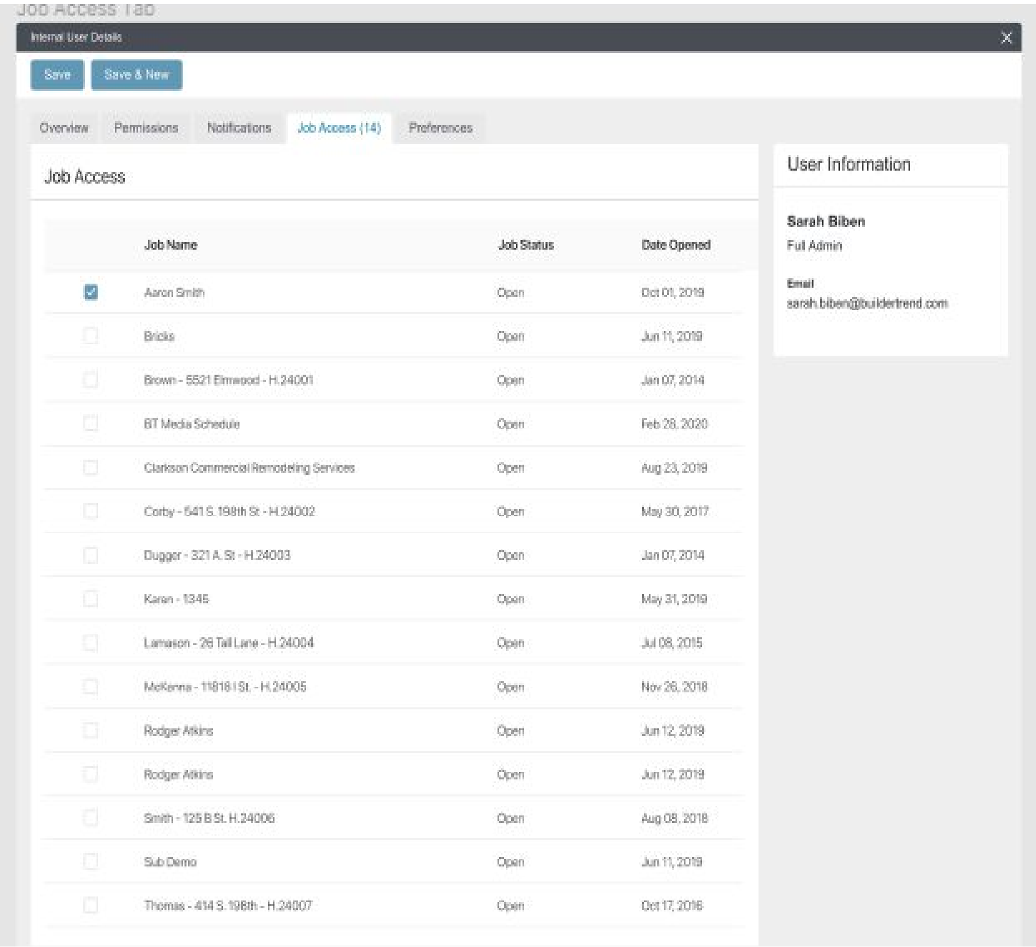
☑ Require location on mobile and track when Ian clocks in or out beyond the geofence.

User Information

Sarah Biben
Full Admin

Email
sarah.biben@buildertrend.com

Figure 7. Current React version of Preferences page that was completed in October of 2020.



Job Access

	Job Name	Job Status	Date Opened
☑	Aaron Smith	Open	Oct 01, 2019
□	Bricks	Open	Jun 11, 2019
□	Brown - 5521 Elmwood - H.24001	Open	Jan 07, 2014
□	BT Media Schedule	Open	Feb 28, 2020
□	Clarkson Commercial Remodeling Services	Open	Aug 23, 2019
□	Corby - 541 S. 198th St - H.24002	Open	May 30, 2017
□	Dugger - 321 A. St - H.24003	Open	Jan 07, 2014
□	Karen - 1345	Open	May 31, 2019
□	Lamason - 26 Tall Lane - H.24004	Open	Jul 08, 2015
□	McKenna - 11818 I St. - H.24005	Open	Nov 26, 2018
□	Rodger Atkins	Open	Jun 12, 2019
□	Rodger Atkins	Open	Jun 12, 2019
□	Smith - 125 B St. H.24006	Open	Aug 08, 2018
□	Sub Demo	Open	Jun 11, 2019
□	Thomas - 414 S. 198th - H.24007	Open	Oct 17, 2016

User Information

Sarah Biben
Full Admin

Email
sarah.biben@buildertrend.com

Figure 8. Current React version of Job Access page that was completed in October of 2020.

Implementation Issues and Challenges:

The challenges our team faced depended largely on the page we were assigned at the time. For example, during the spring semester, one of the challenges our team ran into was needing a component that Buildertrend did not already have. It would have been easy to make the component to solely fit the needs of our page, but Buildertrend wanted any new components to be made into common components. This meant that our component had to be reusable by other Buildertrend employees and pages in the future. This proved to be a challenge as we had to make our component fit our needs, but also flexible and customizable so that it may be used on other pages.

Additionally, when we were working with the Internal User Details pages, we struggled with the Preferences tab the most because it was the one tab that we had to make completely from scratch. Unlike the Data Science Page, which was easier to build *because* we got to make it from scratch, the Preferences tab was very complex and required lots of little components to be made. With our other tabs that we were in charge of, all had some pieces of the puzzle that helped tremendously, but the preferences tab was the most complex and had nothing for it. It was a learning curve and our team had to always be double checking our work with the Buildertrend team.

Another challenge that our team had faced throughout the process was having issues with our virtual machines that were provided to us by Buildertrend. Because of intellectual property agreements, our team was not able to access Buildertrend's code repository outside of the virtual machines we were given. This meant that our team was stuck from making any progress if there were any issues with the VMs, which unfortunately happened often. The virtual machines were riddled with configuration issues that prevented us from having access to the right resources for the majority of the semester. When Buildertrend switched over to a new Source Code Management system during the summer, all of our virtual machines from the past semester were not migrated over accordingly and in result held us back for an excessive amount of time. It took about two months before our team was even able to login to the virtual machines, and even then we were having more problems because of incorrect access permissions, which continued to slow us down.

6. Closing Material

6.1 Conclusion

In conclusion, the purpose of our project was to help provide Buildertrend customers with a faster and more modern user experience by assisting Buildertrend with their process to convert their website from ASP.net pages to React. The goal of this project was to migrate a minimum of 3 webpages to React. Our team accomplished this goal and we were able to convert more than 3 pages. These pages ranged in difficulty level and time needed to complete them.

Over the course of this project our team has learned many things. We learned how to use new frameworks and tools such as React, Formik, Storybook, Ant Design, and Type Script. As both the ISU team and Buildertrend have been working remotely for the majority of the project, we have gained great skills in learning how to work remotely.

Although our project has come to an end, Buildertrend will continue to migrate their website to React in the future using other ISU capstone teams, as well as Buildertrend teams.

6.2 References

We do not have any references.

6.3 Appendices

Appendix 1: Project Takeaways

Our team learned many valuable lessons throughout the duration of our project. We found that there were both benefits and drawbacks of working with a client, and there are a few things we might have done differently if we could begin this project anew. For instance, we may have considered a more creative project. Because our team was working with a client, we were highly restricted when it came to creativity and decision making. Our team did not get as much design experience as we may have had if we had chosen an original project. On the other hand, our team was able to gain valuable, real-world experience that will aid us in our transition from an education setting to the beginning of our careers. We learned that integrating and understanding other people's code is difficult. This is a key motivator to writing clean, well-documented code that will be maintainable and reusable in the future. We learned about the mindset of a business; why Buildertrend switched from .NET to ReactJs, why they wanted to give their customers a better, cleaner, Buildertrend experience.

Additionally, another key learning experience for our team was getting to experience the process of satisfying the requirements set by a client. In most of our classroom settings up to this point, we have been required to build code from scratch, or to design projects that will work in isolation and will require no integration. It is unlikely, however, that our team will be doing this type of coding in the industry, which is why working directly with a client proved to be valuable and worthwhile for our team. Moreover, we learned the importance of communication, not only within our direct team, but also with Buildertrend and its employees. To finish our project and meet the requirements set by Buildertrend, it was extremely important to have meetings often, ask lots of questions, and have our progress reviewed frequently by the React Migration manager.

In the end, this project experience was not perfect. There were benefits and drawbacks, but that is most likely the experience our team will have in the real world. It was good to face the challenges we did so that we could learn from them and have a stronger set of knowledge and skills at the end of our college careers.

6.4 Related Products and Literature

Our team did not have to use any outside resources other than the information Buildertrend gave us.