

Senior Project Defense

Kendall Baker

April 2023

1.) Problem Description

- As a member of the Building Maintenance team at Utah Tech University, we receive a high volume of work orders. However, when it comes to fulfilling a request to fix a hole in a wall, we encounter several obstacles.
- One of the main issues we face is determining the appropriate color and sheen of paint needed for the job. With a large variety of paints in our inventory, it becomes difficult to keep track of them all. Even in cases where a building may have the same paint color, the sheen may vary across different floors.
- Currently, our only method for keeping track of paints is for crew members to make an educated guess or rely on memory, which leads to a considerable waste of time and resources.

2.) Solution Description

- To address the challenges faced by the Building Maintenance team at Utah Tech University in managing their inventory of paints, I developed a web application called TrailBlazerUpKeeper using the MERN stack, which consists of MongoDB, Express, React, and Node.
- The primary focus of the application is to provide a comprehensive view of the rooms in each building and the paints contained within them. To achieve this, I structured the project within containers that includes a form to add a room and an option to view existing rooms. This layout is consistent across all buildings and rooms.
- For example, if a user wants to view the rooms in the SET building, they can click on the icon that displays the rooms in a grid format for easy viewing. In the backend, I created a building details component that allows the application to make a fetch request for buildings and rooms, ensuring that data is displayed accurately and efficiently.

- To create a new room, the user is prompted to input the room number, select the room type from a limited number of options, and then choose the applicable paints, I made a component that allows fetches all the paints in the database and allows the user to select as many paints that are required for said room. The same process applies to creating paints, where the user inputs the paint name, selects the type from a limited number of options, and adds a description of the paint composition. Once a paint is created, it joins the list of existing paints on the application's paint page, making it easy to select for use in creating new rooms.
- Finally, I added a page with helpful links that building maintenance employees can access to streamline their work. Moreover, TrailBlazerUpKeeper's user interface is designed to be intuitive and user-friendly, making it easy for building maintenance crew members to use. The application's grid format for displaying rooms, for example, makes it simple for users to find the room they need and see the associated paints.
- Overall, TrailBlazerUpKeeper offers an efficient and user-friendly solution for managing paints in the rooms across various buildings across campus, saving time and resources for the Building Maintenance team at Utah Tech University.
- I was able to deploy TrailBlazerUpKeeper, I bought the domain trailblazerupkeeper.com to establish a professional online presence for the project. You can view the website if you click here at trailblazerupkeeper.com
- You can also view how i coded my project, I used GitHub to make my code public. You can view it if you click [Here](#)
- I also made a youtube video display my website. You can view it [Here](#)

3.) Technical Overview

- In selecting the technical stack for my senior project, TrailBlazerUpKeeper, I sought to expand my knowledge by incorporating both familiar and new frameworks.
- After careful consideration, I decided to use the MERN stack which includes MongoDB, Express, React, and Node. Given my prior experience with MongoDB, Node, and React at the Utah Tech Code School, I was comfortable using them.
- Although I lacked experience with React, I opted to include it as it is a widely used framework in the industry, and I desired to learn more about it.
- Additionally, I focused on file organization, recognizing that this would improve the efficiency and scalability of my project. Through extensive

research, I concluded that the Model-View-Controller (MVC) file structure was the best fit.

- This structure allowed me to efficiently navigate my code, dividing my project into two sections: the client and the server. Within the client section, I organized my pages and components into subsections, while the server section contained subsections for schema models, controllers, and routes.
- This approach greatly streamlined my project, and allows for me to add to my project which allows for easy maintenance, scalability and readability.
- In the development of my senior project, I sought an efficient way to styling my website. After some research, I decided to incorporate Tailwind CSS into my project, and it proved to be a remarkable tool.
- After familiarizing myself with it, I found it reminiscent of Vuetify which I had previously used in code school, which enabled me to quickly adapt to using it. It was remarkably easy to configure and implement, which significantly simplified the styling process for my web application.
- Tailwind CSS was so easy to use that it allowed me to speed up the development of my website because i wasnt wasting time trying to figure out the styling,this enabled me to efficiently create TrailBlazerUpKeeper, making it a critical component of my project's success.

4.) Research Summary

- From the initialization of my project, I recognized that acquiring knowledge of React was critical as that was a framework I had little knowledge of. As such, I watched a playlist on React from The Net Ninja on YouTube and relied on the official ReactJs documentation to gain an in-depth understanding of the technology. You can view the playlist [Here](#).
- In addition, I consulted a textbook online called, "Pro Mern Stack" by Vasani Subramanian, which provided me with valuable insights and strategies on building a full-stack application and allowed me to see how I could connect my front-end with my back-end seamlessly.
- Throughout my project, I received very valuable feedback from my peers. Notably, Joshua Jerred recommended that I replace the paint text-box with a dropdown menu, improving the flow and usability of the application. Jeremy Hutchings provided critical debugging support, including helping me with debugging the mapping in my building details component and was also a big help with deploying TrailBlazerUpKeeper. Derek Jacketta was also very helpful to me when debugging back-end issues during the early days of my project.

- When deciding on a file structure, I utilized resources such as the MDN website and my prior experience in CS2450 to research and optimize my file structure. you can visit the MDN website [Here](#)
- I utilized the Mongoose library, a popular Node.js library for MongoDB, to create my schema models. The documentation provided by Mongoose allowed me to identify the appropriate data types to use within the schemas, thus greatly enhancing the functionality and performance of TrailBlazerUpKeeper.
- One key aspect I wanted to accomplish was for my website to be user friendly and aesthetically pleasing, so to enhance the user interface and design of my project, I selected Tailwind CSS as the preferred styling framework for TrailBlazerUpKeeper. After researching the most effective methods for integrating Tailwind with React, I utilized a website that provided comprehensive guidance, making the setup process seamless. which you can view [Here](#). The comprehensive documentation provided by Tailwind enabled me to learn Tailwind efficiently.
- while researching hosting websites I decided on using netlify and render.com in order to host the frontend and backend for TrailBlazerUpKeeper because they were free which was a big plus but also they seemed the easiest to use from looking around on their websites.

5.) Future Plans

- I would like to continue working on this project, I have identified the next steps to enhance the functionality and accessibility of TrailBlazerUpKeeper. To achieve this goal, my first priority is to make the platform more mobile-friendly. Currently, the navigation bar and other components such as the icons in the containers do not compress well to the mobile view, which limits its usage potential on mobile.
- In addition to optimizing the mobile experience, I plan to implement authentication to enhance data security. Although TrailBlazerUpKeeper is primarily intended for Utah Tech Employees, it remains publicly accessible as of this moment, increasing the risk of data misuse or unintended data input from unintended users.
- In addition, I aspire to broaden TrailBlazerUpKeeper's use beyond paint tracking to encompass tracking of other building maintenance materials such as carpet squares and ceiling tiles. This expansion will allow for a more comprehensive and versatile solution for the need of the building maintenance department.

6.) Conclusion

- To conclude, this project has been a great learning experience for me as a programmer. It pushed me out of my comfort zone and challenged me to create something that I had no rubric for. While it was a difficult task, I thoroughly enjoyed the process and feel that I have become a better programmer and computer scientist because of it.
- This project has ignited my passion to work on more personal projects that will further test and challenge my skills. I am extremely proud of TrailBlazerUpKeeper and I believe it has the potential to significantly assist the building maintenance crew at Utah Tech University in managing their paints and rooms. Overall, this project has been a valuable experience for me, and I am excited to see how TrailBlazerUpKeeper can benefit the employees for building maintenance.