

Morgan Pritchard

Professor Muvva

INF 654 - Mobile Web Development

December 25, 2024

Project Report

When I initially chose this project, it was a project that was in my backlog of projects that I needed to get done, and this project is for the App Development department at Fort Hays Tech NW. The whole goal behind this project was for our students to be able to post to our social media without the need to give them access to the pages or they have to give us everything that needs to be posted. We wanted them to be in control of their posts. The goal of the project was for students to be able to post images and/or a post and submit it to the instructors for approval, which then would be posted to the page. Due to the nature of the application being internal, the instructors would be the ones who control the students' accounts. With that being said, in the final project, you can create a user (Email, Name, Password, Are they Admin?) but can not delete. There are two logins to test both sides. The posts are individual to each user.

Currently, the website is a fully functional PWA. Built with HTML, CSS, and Javascript, with added CSS framework Materialize to assist in building responsive designs more quickly. I also implemented Firebase, which stores each post that needs approval and some user information that is important for their name and if the user is an administrator. To keep the experience going even when the device is offline, I implemented IndexedDB, a replica of Firebase used when the user is logged in. At the beginning of the project, I set up GitHub Pages deployment that allowed the testing of responsiveness on any website without the need to find my IP address each time I switched networks.

During this project, I immediately understood most of the concepts from my use of Express and Firebase admin, as well as Angular PWA. A significant situation I encountered is that users to be deleted through Firebase must be used through Firebase Admin on a server for the end user's safety. Another challenge I had to consider was syncing the IndexedDB and Firebase when the device came on and offline. To combat that issue, when the application saves data, it first saves it to both IndexedDB and Firebase and an internal array. When the application

goes online, it automatically attempts to resynchronize the databases. The final issue that I encountered was that I wanted to reuse my synchronization code together with users and posts, but due to the complexity of the functions and the synchronization of the databases, I decided to stick with users and posts being split. Adding on to the above about the Github Pages, One issue that popped up is with the use of my GitHub repository as a source control, it has made my project a bit bigger folder structure with the same file names in different folders, I had to make sure that I was not accidentally modifying a previous version of the project. Another issue that I came across was my data was duplicating when I switched back and forth between online and offline mode, it was a tricky bug to catch as it would only happen on Opera GX and no other browser, but if the data was to save, it would upload duplicated data. Once a backend is implemented into this project, progress should be made further due to limitations of Firebase authentication, and posting straight to social media.

In this project, I could go back to the roots of web development with basic HTML, CSS, and Javascript. During this process, I learned that all of the sections of the website need to be organized, or you will start getting lots of different files in one folder. One trick I did, and this project I followed, was using JSDocs to make autocomplete easier. One improvement that I should do is create a new javascript that contains only my objects instead of putting them in the database javascript files. Once time allows, I want to continue with the project, adding a server that will enable full Firebase access to delete and modify users and implement a social media publishing feature. One other feature I would like to implement is that the web application connects to our main department manager system and automatically adds and removes users into the system.