Front Porch is a chatroom platform that lets you connect with friends around the world. Growing up in the south, the front porch was where life happened – friends stopped by, neighbors would gather, and it was the best place to catch the latest gossip. Being far from home, I wanted to recreate that feeling – a place where drop in, catch up, and share what’s new.

From a coding and design standpoint, this was my first self-directed project. Bloc’s apprenticeship program provided user stories and some basic direction on new concepts, like setting up a Firebase database, but mostly I was left to my own devices. Unfortunately, my own devices aren’t that great at design - that’s something I hope to work on improving over time! For now, it’s functional, and that’s close enough for government work while I move on to learn even more skills.

Front Porch was completed using Angular and Firebase (a Google database platform). This was my first time setting up a database, and it took a bit of trial and error to get my data categories correctly nested. This project also included Bootstrap, which allowed for quicker CSS styling and the creation of modals for users to create chatrooms and usernames. I also used cookies to store usernames so that chat participants don’t have to create a new username each time they visit the site. Mmmmmm, cookies…

This project was an uphill battle for me. Near the beginning, I took a weeklong break from coding for family vacation, and getting back into the groove of things while working on an independent project was tough. One of the things that tripped me up was data binding. Because the UI is fairly simple, there’s one primary HTML file, which in theory would have made things easier. However, because the script is broken down four controllers and two services, it was a little tough for me to figure out how to lay out the correct path to follow to get to the data I wanted. I also struggled a bit with understanding the difference between `$scope` and `this`. Once I figured out that `$scope` is a dependency that must be injected, things started going a little smoother.

One of the cool things I figured out was how to use the HTML form formatting to set minimum and maximum character count. I used this for the username modal to ensure that users can’t accidentally enter a blank name, or one that’s too long for the UI to accommodate.

<modal photo>

{% highlight javascript %}

<input required type="text" ng-model="username" class="form-control" pattern=".{3,15}">

{% endhighlight %}

I also enjoyed playing around with database setup. It’s remarkable that there’s a free tool like Firebase that you can use to power all sorts of functional things, like housing messages and room names for a chat app. The approach I took for housing messages was a little different than I expected. I thought that the most logical way would be to group the messages by chatroom. However, with guidance from Bloc’s coursework and my mentor, I wound up setting up message creation so that each message is individually tied to a room ID. So, once the user selects a room, a script runs through the messages database and displays the list of messages, checking for that room ID.

<here’s what the script looks like>

<firebase photo>

<UI photo>

Oh, and another fun thing – using “Bohemian Rhapsody” phrases for my test messages. Paying homage to Freddy Mercury is so much more fun than “test 1”, “test 2”, “test 3”… :)