BlocJams is a Spotify-style music player, and my first front end web development project. This project was designed to take basic JavaScript, HTML, and CSS skills and start putting them into practice, while adding more complexity along the way. The initial work was done in JavaScript, some of the later work was done in JQuery. This was a great exercise to learn the foundations of JavaScript and the importance of persistence – JavaScript is not a skill that comes naturally to me!

BlocJams includes a landing page introducing customers to the product, a “collections” page that shows the albums available for play. At this point there’s just one, but it’s repeated to demonstrate what the UI would look like if more albums were available. When one of the albums is clicked, users are taken to the “album” page with track list to play and pause songs, skip forward and back, and seek bars to control volume and location in song play.

The scripting on the “album” page is quite verbose. Even after introducing some JQuery elements, there are 270 lines of code. As a feature, a play button will display over the song number when hovered and a play/pause button will show in the same place once the song starts playing.

One of the challenges I faced was getting the currently playing song to correctly update in the player bar at the bottom of the page. In the click handler for the “next” and “previous” buttons, I was declaring a variable for the song number using `var songNumber = $(this).attr('data-song-number');` but nothing happened. By adding `parseInt()` to the declaration: `var songNumberCell = parseInt($(this).find('.song-item-number'));`, this allowed the conversion from array location to integer, which allowed the rest of the function to work as expected and correctly update the player song in the player bar.

Overall it’s a nice-looking app. I have to be honest here: the design wasn’t my idea. The image assets and the design specs were provided by Bloc’s apprenticeship program. There’s one design aspect I’d like to change though. I want to tweak the landing page so that the feature list is above the fold. I think the current design has the features below the fold to demonstrate how scripting can make the features animate on scroll. However, as a user I don’t find this helpful because the website’s purpose and offerings are not apparent on page load.

For now, I have a pretty cool, responsive music playing app. Now all I need are some licensing agreements so I can use it play something other than public domain classical music!