Joey Massoni & Maximiliano Castrillon

Project 4: Guitar Central

For this project, Joey and I decided to focus on creating a web application with a built-in database. We came to this decision based on our success developing a different web application earlier in the course called Stickease. Stickease was inspired by task management and work prioritization, yet now, in a completely different direction, Guitar Central is inspired by capitalism and corporate greed. Essentially our goal was to create a simple marketplace offering a great entry into the world of pointy guitars. The shopping platforms behind tech giants Amazon and Apple work well and we saw them as somewhat of an inspiration. Apple's site is clean, simple and has quality sections in your "Bag" that explain exactly what a potential customer is about to purchase, the product's specs and of course, opportunities to get AppleCare and extra accessories. On the other hand, Amazon's online store can seem as massive as the internet itself at times, with the immense inventory it boasts, but at it's core, the shopping cart is functional by showing when you'll receive your products, has sponsored and related items and easily let's you "Save for Later."

We developed the online store in the Cloud9 IDE to be consistent with earlier assignments and labs involving other tools like Firebase and Bootstrap. Firebase, a Google product, is a fast and well connected database that will allow for objects to be stored in the backend of our site, referenced and called upon in different ways. Bootstrap is an open source library of front-end components that will allow our users to navigate around a clutter-free site. Its themes and organization of divs worked well for the online store we wanted.

Once we understood that we would be using Bootstrap and Firebase, we moved on to customize Bootstrap templates like "Pricing" and "Album" in our cart and index files. On our homepage we dropped in images for three different pointy guitars, all of which have an associated model name, cost and a short description. All of this sits over an "Add to Cart" button for each item. Along with customizing these templates and ensuring the layout and information that is displayed makes sense, we initialized Firebase by adding the configuration code block to our index.html page. Next step was to add the app.get paths for each html file on our site: (/, /

cart, /locate, /team). This routing is all part of the express class and will specify a callback function to be called when the function "gets" a request to the route we defined. We're defining the specific endpoints here. Then we had to focus on getting every component to play nicely together and preserving the consistency in regards to links and layout. We worked more on the realtime database which allowed us to store data as JSON objects. In this case, specifically, we were primarily concerned with "name" and "cost" of each tier of pointy guitar, which you'll see in the body labeled as "cheap," "medium" and "high." We were able to implement anonymous user authentication on the home page but didn't implement authentication throughout the site.

In JS we decided to read the contents of the database to the console, separate the individual parts as strings and enter these into an array. Then, elements of the array were called into the appropriate spots from the html. A more efficient way of doing this that would've allowed us to populate the cart on load would've been outputting the data directly from the database and triggering the script .onload. This could've been made possible via authentication that would provide users with their own sections of the database. We would've seen our storefront (Guitar Central) - a key underneath for each anonymous user - and a series of keys representing the items in their cart.

Wrapping up the design of the code, we wanted to make the site as realistic as possible. We went to logojoy.com and created a logo for Guitar Central to be used across the site. Making it realistic meant involving high-res pictures showcasing the products, an about page that explains exactly who you are buying from and a locate page that can show you our physical storefront. Starting with minimal web programming experience, we're proud of where we ended up this semester. With this project we're confident that we achieved what we set out to do: write to and read from our database, creating a minimally functional site with multiple pages and a clean design. It's evident how entire teams and divisions of companies are dedicated solely to creating intricate websites for various needs. Joey's take on the project was that it was fun because of how open it was and the database work was humbling. My take on the project was that making different pieces work together can be gruesome work but rewarding in the end.