EthicCert: Senior Project

1st Curtis, Michael
California Polytechnic State University
San Luis Obispo, USA
mcurti05@calpoly.edu

***Abstract*—This paper outlines the process of creating an Angular webapp to allow scoring of products on ethicmarketplace.us, a Cal Poly startup. The products are scored based on categories of values customer resonate with. This paper will describe these categories and explain the implementation process of this non-technical idea to a technical reality.**

I. Introduction

I decided to make this app because a friend of mine, Garrett Perkins, pitched this senior project at the Pitch2Programmers event last year. I had already been consulting for Garrett with the technical side of his startup, so I knew about his startup and how it functioned already. He described to me his desire for a scoring system for every product on his store. He wanted a way for customers to evaluate the ethical benefits and downsides of each product by looking at some visual representation the summarized the characteristics of a product. So I set about creating an application that facilitated this process of scoring.

A. Ethic Marketplace

Ethic Marketplace is a Cal Poly startup, co-founded by my friend and colleague, Garret Perkins. It was created as an alternative to Amazon for customers who care more about how ethical there products are. By ”ethical”, I mean how sustainable and ”green” they are. All products on Ethic's storefront are in some way ethically sourced, cruelty free, or fair trade. Ethic's storefront is simply a Shopify store, therefore they have no inventory. Ethic is an aggregating site for smaller independent vendors. All payments to Ethic are forwarded to the vendors and Ethic takes a cut. For this reason, the metadata about each product are non-standard and varying. This presented complications with generating standardized data to pipe into my application.

B. Design Paradigms

I went to an art school before high school, so I have an understanding of color and design. However, it wasn't until recently in college where I found that I could apply my eye for art to a technical project. I came across Material Design in college. Material Design is the design philosophy and rules that Google has provided for all it's android apps. Material Design uses shading to create the illusion of depth on a 2D surface. Additionally Material Design posits that there should not be two many colours in any app. Instead, stick to shades of one or two primary and accent colours. I adopted Material Design for this project because:

- Familiarity: I already understand its principles.
- Integration: because I decided to use Angular for my frontend (which is a Google technology), it was easy to use Material for Angular.
- Adaptability: because Material Design is so modular and simple, it is very easy to add my own design changes without messing about with too much of the flow and feel.

II. Solution

My solution is an Angular app. It is a webapp that is designed for desktop use. I have created apps like this before, but never anything using these exact same technologies and having the exact same features. I believe that this project is a testament to my ability to quickly learn new languages and frameworks over a short time period.

A. Technologies Used

My stack consists of three modules. The first module is the customer facing front end. It is the visual representation of all the data I have collected and processed. It is what a customer see when they visit Ethic Marketplace and view a product Originally this was going to be an emedded HTML iframe. It would grab the score from my database and present it in a graphical and stylized manner. In actual implementation, it is an extension of my employee-facing frontend. I found that since I developed the employee-facing frontend first (the dashboard) first, it was easy to add onto it using the same technologies described below.

The second module is the dashboard. The dashboard is what Ethic employees use to assign score to products. It uses Angular which is a MVC frontend framework from Google. I chose angular for two reasons:

- Familiarity: I have used angular before in other projects. There was not sense in learning something completely new considering the vast scope of this project.
- Compatibility: I knew that I wanted to use Firebase for my database. Angular and Firebase are both made by Google so they mesh very well together and there is plenty documentation on how to use them together.

The last module is the backend. My app is serverless, so the backend is cloud functions. All these function do is update my database with the current data from Ethic's Shopify database. Product data is synced between the two and everything else is handled by other modules. This is a very modular way to handles this and I appreciate how simple it is.

B. Approach

My initial approach to creating this application was to create something visual and worry about technical issues later. I have found in my career as a programmer that having a visual representation of my goal provides me with motivation and foresight that I would not have if I only had a console-based proof of concept.

I continued this visually-driven approach until I needed to write some technical code to actually get data. Then it became about writing small proof of concept programs that proved I could retrieve the data I wanted. I knew that integration would be a whole other issue, but I did not worry about it at this time.

Finally, I took every proof of concept program and integrated them into an organized backend and frontend. This was a very involved and difficult process. Once I finally got it, I was very elated. Looking back, writing all the proof of concept programs and then spending the time integrating them may have taken more time then writing the frontend and backend from the start. However, part of my problem-solving process involves breaking programs into smaller complete-able tasks, and I think that this was my way of doing that.

C. Roadblocks

I encountered a number of issues the hindered my progress throughout these past two quarters. I can safely say they had a theme about them. These roadblocks all centered around the integration between and the communication across the modules I created and the technologies I used. I have learned from this project that many libraries are created with unique style guides and conventions that make them difficult to interface with other libraries and APIs.

The first major roadblock I ran across was how to retrieve the product data from Shopify and present it on the dashboard. This was an issue because the Shopify REST API was completely foreign to me. I had no idea which API I need to make and I did not know how to make such a call from my angular app. I found from experimenting, that Shopify used relative cursors for their database. This meant that in case the results of a query were too numerous to be returned in one response, I could not simply ask for:

```
&page=2
```

Instead, I had to make a secondary query with the id of the last product in the list that had been previously returned. This meant my code had to be a low more detailed and generic then I had previously thought. I needed to create a proof of concept to figure out how to chain queries properly.

So I made a quick proof of concept program that requested a page of products and kept requesting for the next page of products until the page returned was empty. It turned out to be different from the solution I actually used, but I think is was valuable practice with the Shopify API.

The second problem I came across was with the severless functions. I knew I needed to make a function the retrieved the data from the Shopify API and put that data into my Firebase database. However, the functions themselves are already request-response type functions. I needed to make a inner request within a function and I did not know how to do this. Eventually, I found out a way to ”await” the return of the inner request before exiting the outer function’s control. Once I had this problem solved, it was only a matter of time before I had a working backend that correctly copied data from Shopify into my Firebase.

These were the main two roadblocks in my project. I, of course, came across other problems, but they were more quickly resolved and did not hinder progress as much as these two. Looking back there are definitely some tasks I could have been working on while I was stuck on these issues. However, I cannot say with certainty that focusing on something else would have helped my solve this issues.

III. Conclusion

This was a vast senior project and the scope was far greater than I initially thought. That is my major takeaway. There were many times when I found myself working on a smaller visual issue instead of working on the bigger issues. Scoping and feature-creep, therefore seem to be my two major pitfalls.

A. Completeness

Initially, I wanted to have the dashboard be able to fully modify product metadata. I have found that I simply did not have time to add this functionality as it is very involved and there are many semantics with how to safely and securely write data to Firebase. I can see myself experimenting with adding this feature after the quarter is over, but as of writing this paper it is not an implemented feature. For now, Ethic employees will have to modify the database directly from within the console. This is not ideal because entering multiline text is all on one line in the console preview window, but it is better than nothing. Tags, however, can be edited in the dashboard. I figured out how to allow for the modification of this data last-minute. You can now edit a tag and change its name, description, and weight. Tag creation itself is not implemented so new tags much be created through Firebase directly.

Additionally, the customer-facing frontend is poorly designed. As of writing this, it is honestly more of a proof of concept than a full module. I have not added any fancy colors or bar graphs. Ideally, I would show this proof of concept to Garrett and if he liked it, I could think about enhancing it this summer. However, I do not have time to fully realize the designs I had for this module currently. It is regrettable since the customer-facing frontend is what most people would see, but I do not think this module shows of the majority of the technical work that went into this project.

The majority of work I have done these past two quarters is displayed in the dashboard and the backend. I believe the backend to be fully implemented and polished. It is very simple and elegant and it took a long time to get right. Because it is so simple, the dashboard operates independently from it. The work that I have done on the dashboard works well because it is does not have a bunch of dependencies.

B. Future Enhancements

In the future, I would like to add the features I promised in my project proposal. I think that it is important to make the tag creation and assignment processes as intuitive and user-friendly as posssible. For this reason, I believe it is necessary to implement these feature if EthicCert is going to be a long term solution for Ethic. It is regrettable that I did not have time to complete this work during the time allotted for this senior project. However, because this project is more than academic, I have a unique opportunity to expand and improve it specifically for Ethic after my senior project is over. I plan to do this and will be in contact with Garrett directly so that we can decide the exact direction of this app.

IV. Acknowledgment

I want to thank Professor Bruno da Silva for being my faculty advisor for this project. The autonomy he gave me allowed me to realize this idea to what I had envisioned. His guidance from our bi-weekly meetings saved me from pitfalls in my development and design process that would have cost me weeks of progress otherwise. I find him to be an excellent mentor with great and relevant experiences to share.