

# “Campaign” Redesign

Daniel Rapuano

## I.) Introduction

### a.) The Project

During my final year at my university studying computer science, we were tasked with creating our own application, from inception to deployment, as part of our graduation requirement. Groups consisted of 3-4 students, and we would have approximately 5 months (December - April) to complete. This process began with pitching the idea to the professor for approval along with our goals for functionality, as well as a detailed timeline of deadlines to achieve those functionality goals. Throughout the semester reports had to be made and presented to showcase progress and that deadlines were being met, which culminated in a final presentation to the computer science department, and a final report detailing our process. That report can be [found here outlining the entirety of our work.](#)

My teammates and I chose an AI focused project. We determined we wanted something to create an application that could in theory be commercially viable (outside of the context of this project); something that would serve a purpose. We decided we would create an application that would allow small-medium sized businesses, that would not ostensibly have a budget for advertising/ marketing, to use emerging AI models to their advantage to generate advertisements for them, and to distribute said ads to social media sites of their choosing. Our project was highly regarded as being one of the best in the department's history, and lauded by professors and students alike for its ingenuity, complexity, and execution.

### b.) The UI

Our project had a deadline of approximately five months to complete. This meant we had to meet our goals which were primarily functionality. However, because the core functionality, page routing, database models, etc. had to be established before any sort of UI could be worked on, this meant being in a constant state of limbo waiting for the backend to be completed in order to actually create any solid user experience. Our application was very function heavy, and therefore UI/UX was essentially all completed at the tail end of the project timeline, giving little time to actually implement it.

While the styling itself was sufficient enough for our purposes, the usability testing that I had performed with a sample size of  $n=10$  consistently came back with the same remarks. It was confusing to use. The sample was chosen through convenience sampling given our lack of time. I could have posted the application online for others to use for feedback, but we had a limit on our API calls, and it would become egregiously expensive to have the general public use the app. I chose those closest (friends and family), who provided broad variance; about an even mix of each gender, ages ranging from 20 - 80 years, and a broad range of experience with technology. Peers within the CS program were not selected due to a bias of being more infatuated with the functionality of the product and how it worked, rather than a natural use-case. Computer Science students were not the target audience for this application, owners of small businesses were, and they presumably would engage with the app in a more natural way.

This application had to be user-friendly in such a way that the target demographic would naturally be able to use it and its features without thought. It was confusing for them, however, and this was not reconciled prior to deployment. The source of confusion stemmed from the file system. While an integral part of our application, and of great use for the user *if need be*, users did not know what “campaign” or “portfolio” meant in the context of the app. Every single user tested, when greeted with the homepage after login did not know how to proceed, typically responding with “what do I do” in some capacity.

## II) Revision

In an attempt to make this application more viable for the target user post graduation, I sought to revise the app to reduce confusion. This meant two things as I could determine. Firstly, let users simply generate an advertisement immediately. This meant bypassing the entire file system entirely, and even the login page if the user wanted. I created a Quick Generate function that granted users to do just so. An updated landing page can be seen in the figure below. The user is immediately greeted with this option, ensuring absolute simplicity.

campAlgn About Devs Login

# Create personalized advertisements with AI

### Promote your product in seconds

Try it without an account. Click below for a quick generation

Quick Generate

Utilizing OpenAI technology, CampAlgn provides an easy and

### Login

Email:

Password:

### Sign Up

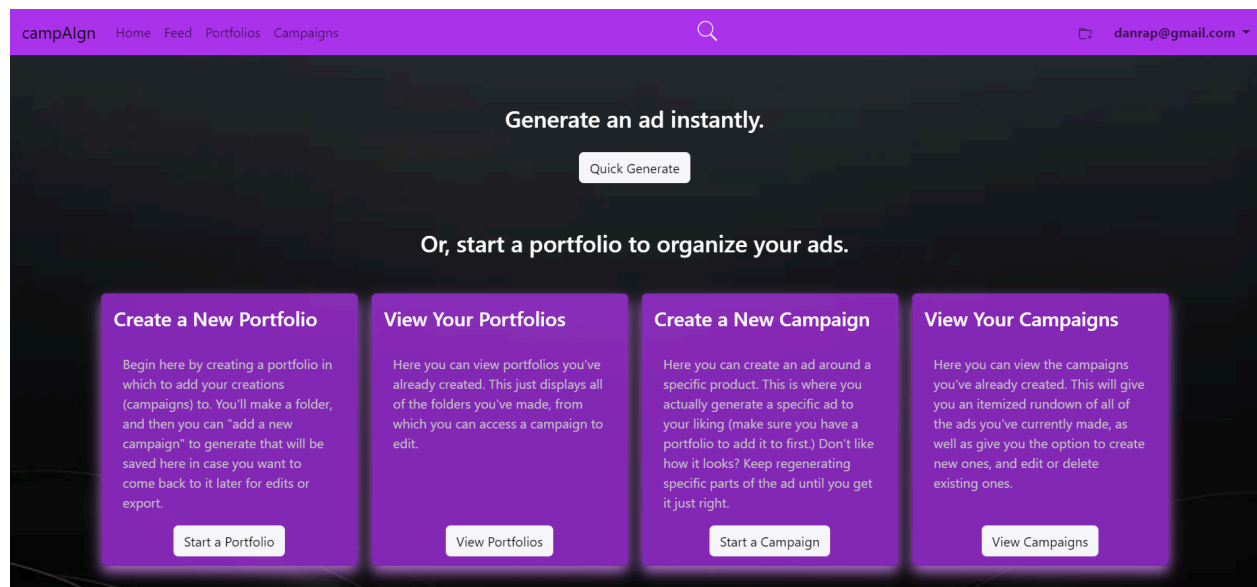
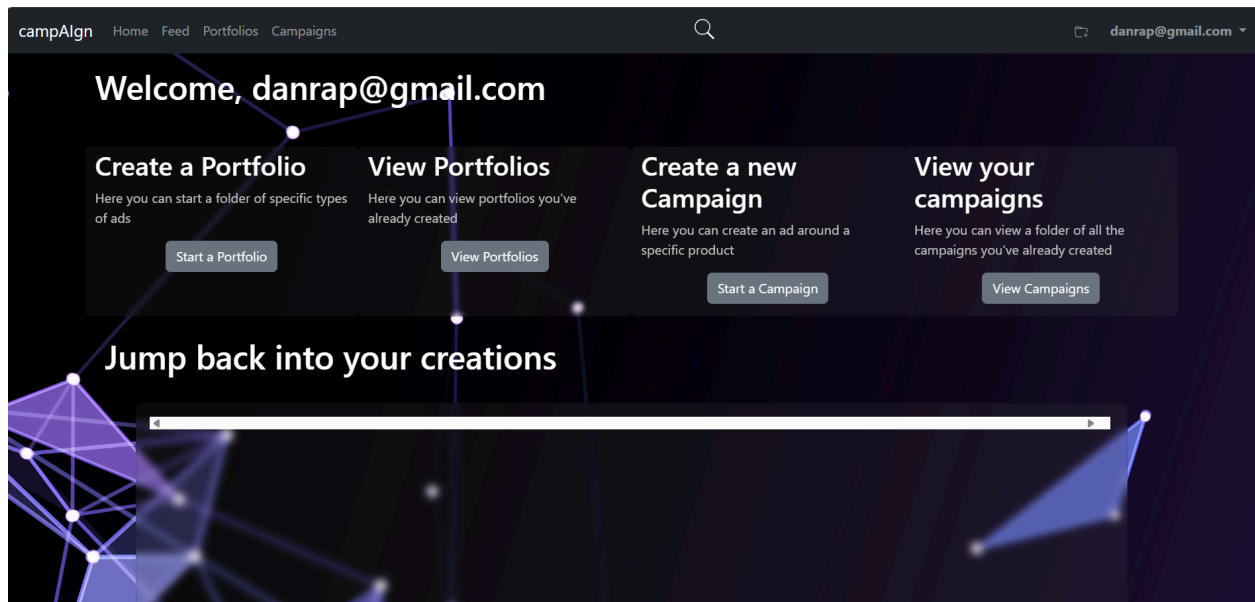
Username:

First Name:

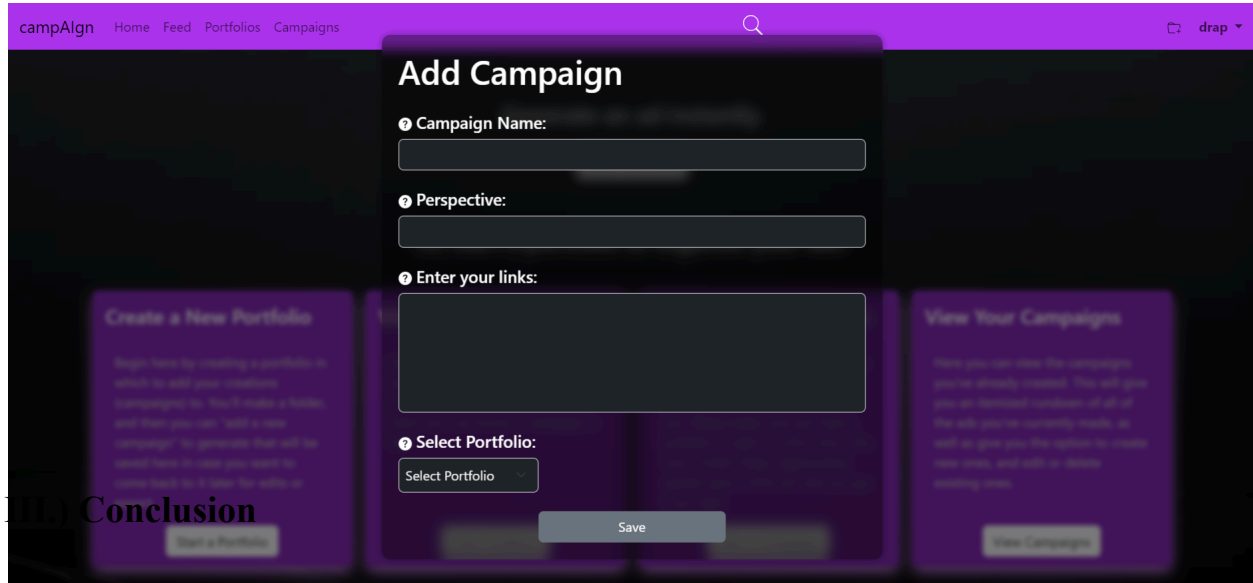
Last Name:

This proved challenging, however, as the generative functionality was linked to the file system, and required the user to establish their destination folders before any generation could be made. I had to recreate this functionality in order for this to work. This feature also displays on the homepage as well if a user has decided to make an account.

Next, I wanted to make sure that if a user did choose to use the file system, it was as straight forward as possible. Without redesigning the entire site, I simply added a more detailed description for each function that appeared on the homepage, where the order of operations on how to use the file-based system appears from left to right. The homepage is shown below; the old version along with the redesign.



To further reduce confusion over functionality for the average user, I additionally implemented basic tooltips to the “add” forms. These now explicitly explain to the user what each form field does and for what purpose, and are denoted by a question mark icon next to the heading of the field.



This application was undeniably successful given the context it was created in, and for its complex functionality. However, with time constraints prevalent, proper consideration for the actual person who would be using it was woefully lacking. Post-graduation I have returned to this application in an attempt to resolve these issues, namely, its obscure flow and file system. I implemented a Quick Generate feature in order to allow users to immediately engage with the core generational component to produce a customizable advertisement to be exported. I additionally added far more explicit signaling to the user for how to use the file system if they intended to, by establishing clearer instructions of the flow, and by adding tooltips for form fields.

In this way, a user could use our application with relative ease, which is essential given that our target demographic are regular people with baseline levels of tech knowledge, who more than likely do not interact with novel web apps. Our site is doing something unique with technology that is relatively new. There is no solid foundational understanding for most people on how these sorts of applications work.

I intend to continue to work on this project through more usability testing, expanding the sample size, and implementing more appealing and more clear styling to signal the most relevant information on the page.