

# Sprint 5 Report

Prepared by: Josue Nunez, Alex Woods, Linhao Yuan

## Intended Progress:

Team DIG's goal for the third sprint was to connect the front end with the backend and to create different dashboard pages to show the analytics for each marketing tool.

Alex's goal for this sprint was to create a report on how users interact with the pages. As well as adding different options for different reports to be made.

Josue: Josue's goal was to merge Alex's work with Google Analytics api to one single flask app that would run both Mailchimp and Google Analytics. Another goal Josue had was to connect the front end with the backend.

Linhao: Linhao's goal was to add pages for other different apis on the front-end, and start implementing the api calls in those pages

## Progress Reflection:

Alex was able to get page statistics for the site, allowing for that to be monitored in the application.

Josue: Josue merged Alex's Google Analytics api code with the mailchimp api under one flask app and incorporated CORS for handling Cross Origin Resource Sharing. Josue connected the front end app with the back end and now runs smoothly displaying json data to the backend. Josue also redesigned the mailchimp json file to only include what is needed for the front end to display the kpi's. Mailchimp api is also designed so that it scans through every email sent in real time updating , grabbing the data and sending it to the front end.

Linhao: Linhao was able to add all the pages needed for the dashboard, but Linhao still needs to implement the api calls into all the pages

## Problems Encountered:

Some problems we encountered were figuring out a way to combine different flask apps into one, we struggled to find the best way to do this to combine mailchimp api code with google analytics api code keeping each one's integrity. We do so using blueprint methodology in flask. A Blueprint object works similarly to a Flask application object, but it is not actually an application. Rather it is a blueprint of how to construct or extend an application. Once we registered each blueprint we were able to combine them into one flask app. We also had trouble testing the backend response data seamlessly as we had to remember each url route name to test for both mailchimp and google analytics. To fix this issue we added Swagger Docs which is a documentation web app that displays all the routes and allows you to test responses for whichever one you want.

## Projected Progress:

Alex's goal for this week is to focus on bug fixes and ensuring that the project can run everywhere with the given readme. We seem to be on track for finishing the project and ensuring it works.

Josue: Josue's goal for this week is to explain some of the changes he made in connecting the front end to the backend, and some packages he needed to make it work. He wants to help development on the front end as the backend is pretty much finished.

Linhao: Linhao will start implementing the api functions in the front end so the front end will display real data from the back end on the dashboard

## Teamwork Reflections:

The past sprint was difficult due to having the thanksgiving break. We were having issues with getting some components of the project working, but were able to get them sorted out. Meetings are an important part of getting the project finished, but we were unable to meet with our sponsor at that time and meetings with the group were delayed due to time conflicts.

## Conclusion:

We seem to be on track with getting everything finished with the project. The last sprint will largely be focusing on ensuring everything works correctly and that we're prepared to present our project. We will help develop more of the front end with the new data being sent to vue.js server and do a lot of error testing. We want to redesign some pages in the dashboard to make the data look cleaner and more effective to bring more value to clients.