Final Report

Prepared by: Josue Nunez, Alex Woods, Linhao Yuan

Abstract

A customer's experience with a brand can be hard to visualize and quantify, especially when there are so many ways of interacting and forming experiences with a brand. That is why Team DIG's Customer Experience Dashboard project intends to provide a view of the average customer experience, collecting data relevant to a brand and displaying that data in an easy to understand dashboard. We gather data via the APIs provided from Google Analytics, Mail Chimp and Campaign Monitor to monitor a user's experience with a site and email champions to get an understanding of how users interact with the brand. The results of this project showed us that it was possible to get a generic overview of a brand and that it was possible to have a comprehensible dashboard that implements a number of different analytics APIs.

Introduction

When first starting the dashboard, users are prompted with a login screen. This is to allow the website statistics to be shown to only the necessary users. After the login screen, users are greeted with the main analytics dashboard. This is the hub from where an admin can go to see the overall user statistics, email campaigns or the export report tab. The user statistics page's goal is to give an admin a general understanding of how users interact with the brand's website. Statistics such as bounce rate, ratio of new to returning users, and session duration aims to aid this. The email campaigns tab next to it, displays statistics such as emails sent, open rate and sub rate to show the successfulness of the email campaigns that are being sent out to customers and users. The last tab is for exporting the reports that are created in the dashboard.

Team DIG: Customer Experience Dashboard

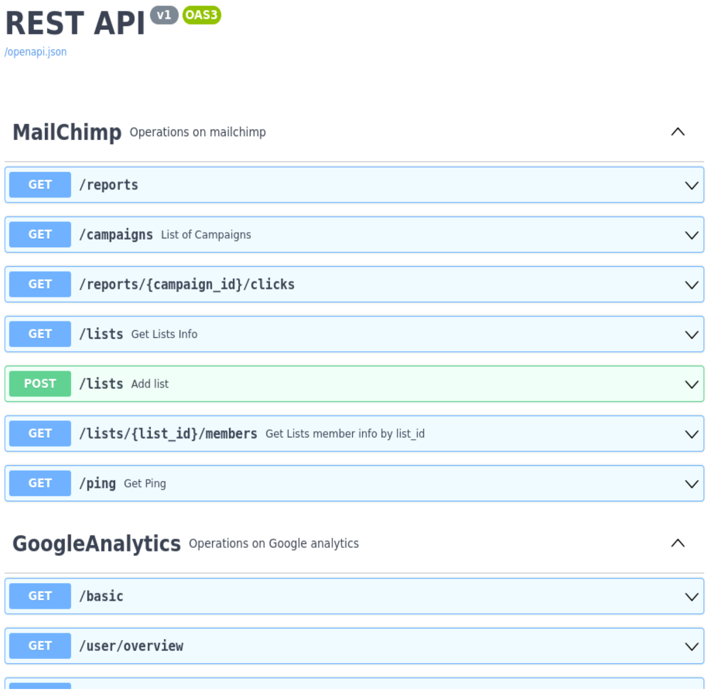
Vue.js:

Vue.js features an incrementally adaptable architecture that focuses on declarative rendering and component composition. The core library is focused on the view layer only. Advanced features required for complex applications such as routing, state management and build tooling are offered via officially maintained supporting libraries and packages.

Vue.js allows for extending HTML with HTML attributes called directives. The directives offer functionality to HTML applications, and come as either built-in or user defined directives.

Flask

Flask is a great option for developing RESTful APIs and is very scalable. Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. It has a very lightweight Web Server Gateway Interface (WSGI). Flask stands out for its easy-to-extend philosophy. It was designed to scale up to complex applications and to support an easy and quick start. Another great advantage of Flask is its functionality. Even though it offers suggestions, Flask does not mandatorily require project layouts or other dependencies. Instead, it allows developers to choose the libraries and tools they wish to use and additionally has various extensions available that are provided by the community. For our project we use snowflake and Cors in our flask app to help send data from the back end to front end seamlessly. We also incorporated the use of blueprints which Flask uses for making application components and supporting common patterns within an application or across applications. Blueprints can greatly simplify how large applications work and provide a central means for Flask extensions to register operations on applications. We separated our Mailchimp API and Google Analytics by blueprints and then combined them together under one app with the help of blueprints. Flask is also very useful for being able to see live changes and responses to your code on the server during run time. This makes it very easy to change and fix errors in our code on the fly. We also incorporated the use of Swagger Docs to be able to document the routes and Get requests we created and the data they returned under one page. This was very useful for testing and checking for errors in the API’s.



REST API v1 OAS3

/openapi.json

MailChimp Operations on mailchimp

GET /reports

GET /campaigns List of Campaigns

GET /reports/{campaign_id}/clicks

GET /lists Get Lists Info

POST /lists Add list

GET /lists/{list_id}/members Get Lists member info by list_id

GET /ping Get Ping

GoogleAnalytics Operations on Google analytics

GET /basic

GET /user/overview

Mailchimp

Mailchimp is a marketing automation platform designed and developed for businesses using email to reach out to their target markets. Email marketing is an effective way for organizations to run extensive and worldwide promotions without stressing their marketing resources. Despite the popularity of social media marketing, email is still an effective way to promote business, acquire new customers and retain the existing ones. Many stats and figures prove that email possesses a lot of marketing potential, and businesses should never overlook it as a viable means for expanding the consumer base, strengthening the brand, and improving revenues. Dig Marketing has found great success in using email marketing campaigns to track user analytics such as bounce rate, click performances, open rates, revenue made after clicks. Mailchimp has a Mailchimp API for developers to use to help send and receive live data. This is very useful to track important data a client might want to see in a single dashboard platform. This allows for DIG to not have to share screenshots or emails of data every time because it is automated and only includes client data and no other sensitive data.

Google Analytics

Google Analytics is a powerful statistical tool that allows you to track users and traffic on your website. Most users interact with it with the Google Analytics Dashboard, but for our project we used the Google Analytics API for tracking user data on our website. We created different reports via different API calls and were able to view data for the different requests we'd made. This allowed DIG to utilize the data for our Dashboard and properly query the reports we wanted to make.

Areas of Technical Growth

Josue Nunez:
Prior to this class I have never programmed a web application in Python and also have never used the Flask Framework. I've learned so much about the Python Language and its benefits in creating web applications. For instance, it helped me realize the benefits of Dynamic Typing. I noticed an ease in writability for coding because we don't have to declare the type of variable while assigning a value to a variable in Python. Also, not having to build and compile each time I want to run was very beneficial when trying to do quick fixes in my code and minor changes to the output. This allowed me to test my code quicker and program and API faster. It's been a while since I have programmed a client server communication based program and it was very useful knowing how to grab data from an API and send it from the backend to the front end by using different connecting methodologies like CORS to be able to communicate between servers. I think this will serve me in the long run in my career path as I will be pursuing Data Analytics. Using Github to switch between branches, pull, push, and merge requests were used a lot during our project and it was a nice refresher on how Github can be a great platform to collaborate on a coding project and one that is used a lot in the real world. So I am very grateful to have had a project that used Github as well as Zenhub. Overall, I learned a lot about dynamically typed languages, creating a web app, client/server communication, and learning how to communicate well with my team to have the most efficient and productive outcome.

Alex Woods:
Prior to this semester, I had no experience building web applications using Vue and Flask. I had some experience building basic static websites and experience with python, but not to this capacity before. For my part in designing and constructing the backend of the project, I learned some of the utilities that Flask offers for working with data. This allowed me to create an effective backend that could connect with the rest of the dashboard.

Coming into this project, I had only used Google Analytics' dashboard as a user in a company and had no prior knowledge of the API that Google Analytics offered. Learning ways to implement this API in our project was interesting. I was able to gain a great understanding of how data from Google Analytics could be used to track data and statistics on a site, as well as, gaining ideas on future projects. Overall, I'm glad I was able to work with this portion of the project.

Lastly, I wasn't familiar with creating reports on and planning goals around sprints. It was a great skill to develop and it seems as though I'll be using it for all future projects. I feel that my technical writing skills have improved because of this as well.

Linhao Yuan:
Before working on this semester long project, I already had some experience with the front-end development from my internship, but I was only a intern working on pre existing projects. This project gave me an opportunity to built everything from scratch, also since I am the only on working on the front-end for this project, I also had to make a lot of decisions on my own, it definitely gave me a taste of what its like working in the real industry as a developer. Also, when I gained lot of experience on how to communicate with the back-end developers who are working on the same project. And of course, through this project, I was able to learn my new knowledge on the front-end development that I didn't know before, these experiences can be beneficial for my future career.
Overall I am very happy that we are able to making this project working as we planned in the beginning of the semester.

8 Ethics Principles

Principle 1: PUBLIC

1.01. Accept full responsibility for their own work.

We acknowledge the contributions from each individual are documented in github and the reports.

1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.

For our project, we aimed to only draw data that is relevant to the project and not infringe on an individual’s privacy.

Principle 3: PRODUCT

3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.

Goals for our project were laid out before each sprint to ensure that we stayed on track with meeting them. If the goals exceeded the sprint they were addressed and evaluated to fit in the next sprint.

3.11. Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.

Documentation for the project was recorded in the sprint reports of this project. This included the issues, setbacks and goals that were achieved during the sprint.

3.13. Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.

Data that was gathered was obtained from ethical, lawful sources and was only used for the essential functions of our project. Data that was collected wasn’t used for any reason that wasn’t in the project’s thesis.

Principle 7: COLLEAGUES

7.03. Credit fully the work of others and refrain from taking undue credit.

The work from each individual was documented in github and in the reports. We each acknowledged the work other team members did and gave credit where it was due.

Principle 8: SELF

8.03. Improve their ability to produce accurate, informative, and well-written documentation.

The project included reports, comments in the code, and readmes to ensure that there was accurate, well written documentation. The goal was to ensure that readers could understand what was happening in each part of the project.

8.04. Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.

Each part of the project had time devoted to understanding how the software and environments worked. Readmes were also included in the documentation to ensure that other users could understand the section of the project.

Teamwork Reflections

At the start of each sprint we'd meet to discuss the goals of the coming weeks, as well as, the issues we may encounter. We also stayed in contact via discord, email and through the notes on github. This allowed us to have reasonably good communication within the team and with our sponsor during the duration of the project and allowed us to not encounter any large issues along the way. Halfway through the project, we changed contacts with our sponsor company and we weren't able to ensure the same level of communication that we'd had before. Despite this, we continued to have our weekly meetings to cover where the project was and the direction we'd planned to take with it. Each team member focused on one aspect of the project to divide work and ensure that it was getting done properly. For instance, Linhao focused on the front end, while Josue and Alex focused on getting data in the backend. This allowed us to work on our parts without getting in each other's ways. Overall, we succeeded in having both good communication and teamwork as a group.

Conclusion

This semester, our team was able to develop a functional web application that tracks a user's experience. Utilizing different APIs and frameworks, we were able to create a dashboard that tracks users and gives an invaluable overview of a brand. This project is best seen as a proof of concept and can demonstrate the possibilities of creating a larger dashboard that can get an even larger view of a customer's experience with a brand.

Major Challenges Encountered

Our team faced a number of challenges throughout the development process. As a team, we didn't have a lot of experience building web applications with Flask or the different API that we'd used throughout the project. This caused the sprint planning to be less accurate than what we'd counted on.

We had originally planned to incorporate user demographic data into our project, however we ran into issues with the way data is collected in Google Analytics. Google Analytics doesn't allow for users to use a demo account for their API's and the website we were using didn't track the demographic data for the users. Because of this, we weren't able to add that to our project.

Originally, we had planned on using Facebook's API to gather data from Facebook. We eventually scrapped the idea because of time limitations and learning we could get most of our data needs from Google Analytics instead.

Connecting the data from Flask to our dashboard was more challenging than we'd expected. None of us had experience with this in the past, so we all had to learn how to work with it.

Future Work

For further development of this project, there are a few things that need to be developed upon. First, we have a login screen but no database to support it. We had originally planned on creating one, but ran out of time to implement one. Second would be to implement an easy means of adding new brands and websites to the dashboard. We were largely limited to only having access to DIG’s website for Google Analytics’ statistical data. If we’d like to add more sites to get data from we’d need access for testing and fail safes for when a site doesn’t record a specific statistic, such as DIG not recording demographic data. Due to time limitations and access to different websites, we’d need to expand on those limitations in the future.

Keys to Success

Our team feels that our commitment to weekly meetings, strong communication, and our own personal skills we brought with us to the team allowed us to succeed. Many of the goals for our project expanded upon the skills that we entered the class with. The prior knowledge that we’d gained proved to be invaluable for overcoming the issues that we had dealt with. Our weekly meetings allowed us to bounce ideas off of each other and expand upon them. This proved to be integral to overcoming the challenges for this project. To conclude our statement in our Teamwork Reflection: a successful capstone project relies on organized planning, up-front open communication and a team with diverse skills.