Team DIG

Alexander Woods
Josue Nunez
Linhao Yuan

Supervised By: Joe Royston

Table of Contents

Overview

The Customer Experience Dashboard will connect different API's and show a particular brand's site analytics, marketing, and other data in near-real time. Using Dashboard-type visuals, this program will help visualize the omni-channel customer experience for a brand.

Languages

The Customer Experience Dashboard will be written in Python for interacting with the different API's that we'll use to get the data. We'll be using JS to create the frontend for our project.

Frameworks, Libraries, and API's

Google Analytics

Google Analytics will be used to track statistics regarding the users on the website. This will include statistics such as bounce rate, users and new users, average session duration and sources for incoming traffic.

Google Target Search

Google Target Search will be used to track statistics regarding the demographic of the audience.

Facebook Ads

Facebook Ads will track the statistics of users and advertisements specifically from facebook and instagram.

Campaign Monitor and MailChimp

Campaign Monitor and MailChimp will be responsible for tracking marketing with automated email campaigns.

iDonate and Donorbox

iDonate and Donorbox will be used to track the statistics regarding donations and the users from each respective website.

Vue.js

We will be using The Vue.js Framework to create the front end of our application.

Code Repository Organization

Branching

Our git workflow is composed of 3 branches.

Master

This branch will contain the most stable version of the code for each sprint. All pull requests will be thoroughly reviewed before they are merged.

Development
This branch will contain the latest version of code that is currently being developed.

Documentation
This branch is reserved for all work related documentation of the project.

ZenHub
Our group will be using the following columns in Zenhub for organizing our project.

Icebox
Low Priority Issues that will be addressed in the future

Sprint Backlog
The backlog will contain issues that are in the current sprint, but that haven't been worked on yet.

In Progress
This column will contain issues that are currently being worked on.

Review
This column will contain issues that have been completed, but not yet reviewed

Done
Done will contain all issues in the current sprint that have been completed, reviewed, and approved.

Closed
This column will contain all issues that have been completed and are not part of the current sprint.

System Organization
The Customer Experience Dashboard users will be required to login as a verified user for a company. We will be using a 3rd party application connected to the software login that verifies an email is an official email associated with the company and organization allowing you add certain access to authorized individuals as the company sees fit. We won't allow for guest users, as the data we'll be pulling from the API's is vulnerable.

Users will be prompted in the dashboard to view different analytics drawn from each of the API's.

This will contain graphs and charts visualized in a way that shows the data in an easy to understand way.

The service layer interface will be responsible for interacting with the different API's on the backend. In the dashboard, the user will request data from a list of options such as number of users on the website and the data will be fetched from the respective API. The primary service that our app will provide is to link all of the different API's under one dashboard.

In regards to storing data, we won't have a database to store the data that we pull from each of the APIs. Instead the data will be sent directly to the user after being requested. This will be utilized to cut down on the cost and complexity of the project.

Non-trivial Requirement

The two Non-trivial requirement will be the front-end and the back-end, the sponsor did mentioned that they are not too worried about the front-end, they want us to focus on the back-end.

Front-end:
As of right now, we still haven't decided what we want to do with the front end, as we mentioned, the sponsor wants us to focus on the back-end, so we will decide on the front-end based on what information we are able to feed through the API.

Back-end:
We will be taking data from multiple sources, they all have their own API, we will need to find a way to access those, put them all together and establish our own API. We will also need to do some conversion with those data since our client does want more detailed information when they access our application.

Mockups

The user will see the login screen when they enter the app, once they login, they will be able to see visualized reports of their paid search, targeted advertising, email marketing campaigns each month. The clients will also be able to access detailed analyses of the report.

Work Policies

With how our sprints are laid out, each one of us will be targeting one specific part of the project at a time, but with each task one of us will focus heavily on it. Currently we're planning on having Linhao work on the front end. He mentioned that he had experience working with it in the past. For the backend, we plan on having Alex and Josue work on getting the required data from the APIs.

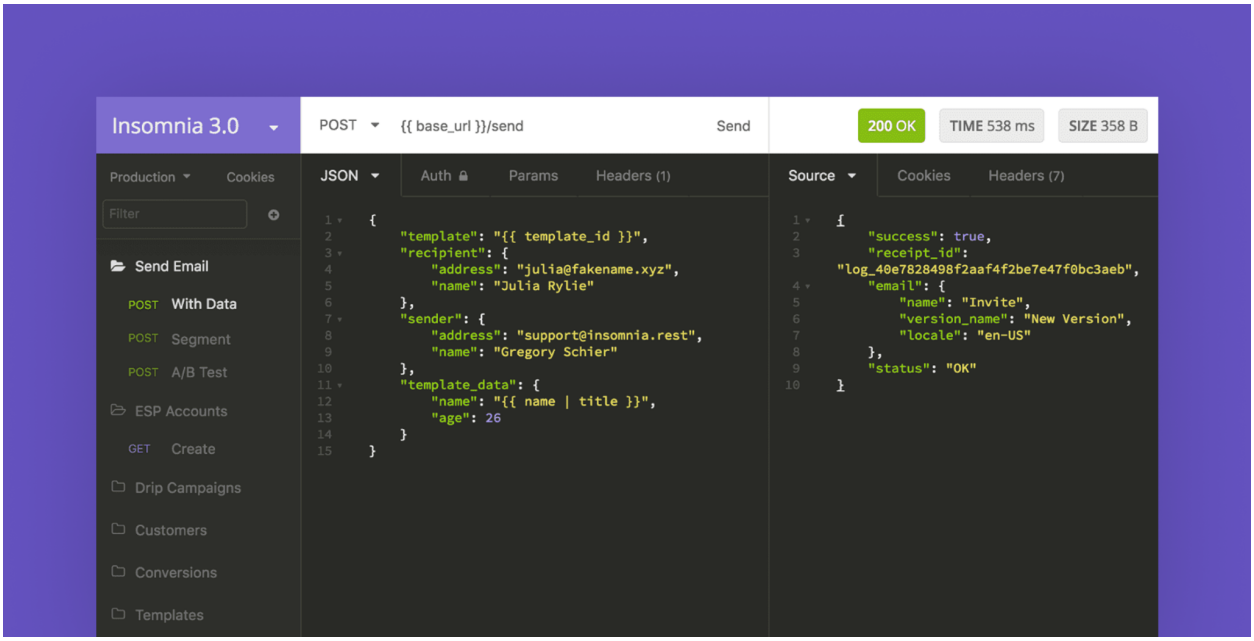
Work Environment and Testing

For our work environment we will be using Visual Studios 2022 adding the necessary plugins and extensions for the environment to work seamlessly with Python and Javascript.

We will be using insomnia to test api connection, the sponsor wants us to focus on the back end, so we will be focusing on the api connections for the majority part of our development. We will be doing our web development testing with Unit.Js which is an assertion library for javascript to help run tests as well as Vue Tests for Unit and Component testing.

- **Unit:** Checks that inputs to a given function, class, or composable are producing the expected output or side effects.

- **Component:** Checks that your component mounts, renders, can be interacted with, and behaves as expected.
- **End-to-end:** Checks features that span multiple pages and make real network requests against our Vue application. These tests often involve standing up a database or other backend.



Deliverance and Milestones

Sprint 1

- Figure out how each API plugs works
- Starting establishing the API for back end*

Sprint 2

- Establishing API

Sprint 3

- Front-end
- Connect Front/Back End

Sprint 4

- Polish Front-End
- Bugs

Sprint 5

- Documentation / Polishing

Sprint 6

- Presentation and product submission

Feasibility Study Grading Rubric (100)

- Technical Content
    - Software architecture / organization (10)
    - Rationale for selected approach (5)
    - Discussion of languages / frameworks / APIs used (5)
    - User experience / interface considerations (5)
    - Data storage / deployment considerations (5)
    - Evidence that required software has been installed, prototypes built, etc. (15)
    - Test Plan (5)
- Team Considerations
    - Division of labor / work environment identified (10)
- Sprint Planning
    - Deliverables / milestones clearly identified (10)
- Document Presentation
    - Proper use of figures / graphs / charts (5)
    - Professional appearance / formatting / overall quality (5)
    - Time tracking Up-To-Date (10)
- Presence of confidential peer assessments to-date (10)