Project Requirements Document v1

Company: CJ Affiliate

Project title: The Great, Fantastic Project

Team: The Great, Fantastic Team

Team Lead: Joey Zhang

Team Scribe: Nathan Guan

Team Members and Emails:

- Joey Zhang (chongyu@ucsb.edu)
- Nathan Guan (nguan@ucsb.edu)
- Kyle Ng (kyleng@ucsb.edu)
- Minliao Li (minliao@ucsb.edu)
- Zhuo Chen (zhuo@ucsb.edu)

Background:

Today’s affiliate business sees a world blooming with opportunities, millions of advertisement offers lack the means to connect with a provider while the providers struggle for a perfect fit. The providers don't have to be professional enterprises. Take the example of home bloggers. Home bloggers blog about something they are passionate about and they want to share their passion with their readers. For them to recommend products, they would need to conduct hours and hours of research. Not only that, new products are added to the market every day and prices are always in flux. There is currently no way for the blogger to do all this research in a timely manner and still provide the best offers/products to their loyal readers.

Goal:

The goal of this project is to provide a way for bloggers to recommend deals and/or projects to their readers based on their blog topic(s). With this product, they input keywords from their blog that they want offers for. From there, our web application will supply offers based on those keywords to the blogger in the form of a list. The blogger
will be able to select which offers they want to post onto their site through a HTML snippet.

**Input:**

The input of the web application will be selected from a list of keywords. Once selected, the algorithm will take into account the current trends from an external source (Twitter for example) and produce an improved input using them, it will create a query for our backend server.

**Output:**

The output on the web application will be a list of offers best suited for the user based on their selected keywords. This output will most likely contain multiple offers for any combination of keywords selected. From the offers displayed to the user, one will be able to select however many of the offers that they want to use for their blog. The process for getting the offers a user wants onto their blog will be easy because they will be able to get an HTML snippet from our website that can be pasted onto their blog. This HTML snippet will then display the relevant offer on their blog once embedded in said blog.

**Procedure:**

We will produce an algorithm that will take in sample data from CJ and filter out the irrelevant products and offers. Then we'll pass these filtered results into another database which is ready for presenting to the user. Then we will display all of the offers fitting the user's need (the user’s need is defined by their inputted or selected keywords at this stage) on a separate web application. We will also build an algorithm which promotes ads based on our filter (in this case, our filter is the trending on social media). This will allow bloggers to advertise ads that's interesting to the viewers, maximizing the profit they may achieve through their blogs.

**Milestones:**

- MVP: Web app with a display of “good” offers
- Algorithm that uses keywords as input and output data with those key works in the description
- Takes in sample data instead of hard-coded data
- Search bar for list of keywords
- Selectable keywords for blogger to use
- Buttons to start the sorting process
- Output HTML/JavaScript code for the blogger to post on their blog

Technologies we plan to use:

- Heroku
- Code from CJ
- Frontend: HTML, CSS, JavaScript, ReactJS
- Backend: Java, Python
- Natural Language Process API

Diagram:
User Cases:

1. As a blogger, I click open the weblink, then I want to choose from tags from lots of different categories, so I can get the offers that fit my content best.
   **Test:** I select some tags for my blog, I get the advertisement that fits my blog best.

2. As a blogger, I want to sort my offers by features, so I can choose from offers easier.
   **Test:** I select a sorting criteria, The offer displayed on the web changes in correspondence.

3. As a blogger, I want to get access to more details concerning the AD I’m looking at from the link provided, so I can decide whether it’s a fit for my webpage.
   **Test:** I click the link of the ad, then it opens a redirected page which contains the details of the ad.

4. As a blogger, I want to search from the keywords, so I can find the keywords relevant to me.
   **Test:** I click on the keywords bar, which expands to show me more keywords, I can also click on a search button which gives me a search bar.

5. As a developer, I want to store my data online, so I can provide the information to the bloggers.
   **Test:** We setup a free online database, and we can access this information through an API.

6. As a developer, I want to have test tools and test cases so I can tune my program and see if they are working as expected.
   **Test:** We develop a set of tests that features the minimum running environment of our program.

7. As a blogger, I want to get random offers, so I can easily get one offer if I don’t have a preference.
   **Test:** Clicking “get random offer” and getting new offers every time.

8. As a blogger, I want to view the trending topics, so I can get a sense of what is popular.
   **Test:** I can see the trending topics in the webapp listed on the side, which is gathered by our algorithm and updated regularly.

9. As a blogger, I want to email the AD link to others by one click, so I can share the AD easily.
   **Test:** I click a button after putting in an email address, then it redirects me to the email sending page.
10. As a developer, I want to keep track of how many people have viewed my webapp, so I can know how I am doing.

**Test:** I see the change in the number of people if I look at a certain page.

**Overview:**

- **Sprint 1:**
  - Choose tools or APIs to use
  - Outline workflow
  - Work on front end web app and tool to read in offers (separate for now)
- **Sprint 2:**
  - Continue working getting web app up and running
  - Work on simple algorithm that takes in keyword and output data
  - Combine tools with web application front end
  - PRDv1 release
  - Work on test cases
  - Get database up and running
- **Sprint 3:**
  - Work on tool to sort relevancy of offers
  - Improve algorithm with the aforementioned tool
  - Improve web app with better UI
  - Develop more test cases
- **Sprint 4:**
  - Assure bugs are fixed and MVP is ready for short demo
  - Test cases pass
  - All hard coded data is completely removed

**Github Commit link:**

https://github.com/minliaoli/CJ-Affiliate-Project

**Github Issues link:**

https://github.com/minliaoli/CJ-Affiliate-Project/issues