Category: Technical Test
Position: Ruby/Rails Developer

Overview

- Create a web application that will extract large amounts of data from the Google search results page.

  You need to find a way to work around the limitations of mass-searching keywords as Google prevents it. Be creative.

- Store this data and both display and report back this data to users.
- Users must be authenticated to use the application.

Application Requirements

1. Authenticated users can upload a CSV file of keywords. This upload file can be in any size from 1 to 100 keywords.

2. The uploaded file contains keywords, each of those keywords will be used to search on http://www.google.com and they will start to run as soon as they're uploaded.

3. For each search result/keyword result page on Google, you will store the following information on the first page of results:

   a. Total number of AdWords advertisers on the page.
   b. Total number of links (all of them) on the page
   c. Total of search results for this keyword e.g. About *21,600,000* results (0.42 seconds)
   d. HTML code of the page/cache of the page.

4. Allow users to view the list of their uploaded keywords. For each keyword, users can also view the search result information stored in the database.

All features must have both a Web user interface. If you feel like going the extra mile, you can also add an API to the application (optional). Refer to the below sections for a detailed description.

Web UI

The following screens must be implemented:

- Sign in.
- Sign up.
- Upload a keyword file.
- View list of keywords.
- View the search result information for each keyword.
- Search across all reports.

API (optional)

The following endpoints must be implemented:

- Sign in.
- Get the list of keywords.
- Upload a keyword file.
- Get the search result information for each keyword.

Users would need to sign up via the Web UI (in-browser) to use the API.

Technical Requirements

- Use Ruby on Rails (7.x.x).
- Use PostgreSQL.
- For the interface, front-end frameworks such as Bootstrap, Tailwind or Foundation can be used. Use SASS as the CSS preprocessor.
  Extra points will be provided to the neatness of the frontend.
- Use Git during the development process. Push to a public repository on Github or Gitlab. Make regular commits and merge code using pull requests.
- Write tests using your framework of choice.
- *Optional*: deploy the application to a cloud provider e.g. Heroku, AWS, Google Cloud or Digital Ocean.