Screening Test

Tasks

1. Download 50 public profile PDFs of your connections (randomly) from LinkedIn.
2. Extract text from the above PDFs and store them in a CSV.
3. For every profile data (text), find out the most frequent words and essential words used. It shouldn't contain stop words (like is, the, an, etc.).
4. Create two web APIs using flask/Django or another framework of your choice.
    a. The first web API should take a PDF file as input and return the text in it in JSON format.
    b. The second web API should take text data as input and return the most frequent words and important words (as mentioned in 3) in JSON format.

Details

Note: ***We want you to explore the tasks on your own. Search on google or any other search engine of your choice to understand how you're going to perform the tasks. However, if you don't understand anything or want hints, ask the questions in the discord community.***

Task 1

You can download the public profiles of others manually from their profile page. Save all the PDFs in a folder.

Task 2

You have to write the code to extract all the text data from PDFs and store them in a CSV. Simple python code should do the work.

Task 3

First of all, find out how to remove the stopwords from the data that you've created. The algorithms for frequent and essential words are your territory.

Task 4

This task is straightforward for those who have a bit of hands-on developing web applications. Most of the assignment is done in the previous tasks. You just need to make APIs out of it.

Timeline

The technical marathon starts from 24th August 2020 and ends on 28th August 2020 (5 days).

Notes:

The time you give in is flexible. You just need to make sure that you try to complete as many tasks as possible in the provided timeline.

It is always a good idea to be active throughout discord. Download the discord for your desktop and mobile phone to get more hints and help.

Assessment

If you don't have a GitHub username, then it's high time that you should make one! Each of the four tasks shall have a specific score, and the aggregate score of all the four tasks shall be 100.

The most critical parameters while scoring is:

1. Frequency of commits on GitHub (use git CLI).
2. Quality of your code.
3. The way you demonstrate your solution.

4. How well you maintain your documentation. The scoring on each task shall be done unbiased without considering the candidate's educational or professional background. We will evaluate based on what you submit at the end!