Candidate Evaluation Assignment for Backend Developer

You have two weeks to complete this assignment. Please read the assignment and follow instructions to develop your own application as required:

Introduction:

You will download 3 ebooks mentioned below and parse the text. You will then perform text analytics tasks on your data and store the results in your DB. You will then build an API to serve those results on request.

In Summary, the assignment comprises of:

1. Parsing text
2. Applying text analytics to the data
3. Storage of data
4. Serving the results over an API

Description:

- For the purpose of this assignment you can choose a language of your choice out of the three:

1. Python
2. Java
3. Scala

- You will download and use the following ebooks:

Please download from the "Plain Text UTF-8" Link

1. The Notebooks of Leonardo Da Vinci : http://www.gutenberg.org/ebooks/5000
2. The Outline of Science, Vol. 1 (of 4) by J. Arthur Thomson : http://www.gutenberg.org/ebooks/20417
3. Ulysses by James Joyce : http://www.gutenberg.org/ebooks/4300

- Text Analytics:

1. **Word Count**: First you will find out the total number of concurrences of each word in each document. For example (Word : "Where", Count : 42, Word : "Help", Count : 12). You should treat all uppercase and lowercase words the same, example (If "The" and "the" occur in a document then the total count of word "the" will be 2). Feel free to use any algorithm/programming techniques/patterns to achieve the results. *Do Not* use a module or library for this task.

2. **Parts of Speech (PoS) Count:** (Nouns, Verbs only) Next you will find out the total counts of Nouns and Verbs in each document. This means that your application will (A) Find and store all the Nouns and Verbs in a document and then (B) Find and Store the total number of Nouns and Verbs in each document. Example- A: (Word : "Apple", Count : "7") - B: (Nouns : "54463", Document : "The Notebooks of Leonardo Da Vinci"), etc. We recommend using NLTK (https://www.nltk.org/), Stanford CoreNLP (https://nlp.stanford.edu/software/index.shtml) or CLIPS-Patterns (https://www.clips.uantwerpen.be/pattern) though you are free to choose any other library or write your own implementation for Noun - Verb extraction
3. **Sentence Difference *(Bonus)*:** Here you will take each sentence in a document and compute its difference with every other sentence in each document. You can choose on any metric/technique you may want to employ to compute the difference. In addition, your application should be capable of computing the difference between a sentence input by the user, and all other sentences in each document.

- Storage:

You must store all computed metrics from the last section in a database. You can choose between:

1. mySQL
2. Mongo DB.

- Application Programming Interface:

Note: *Please design, specify and Document all the parameters and specifications for the mentioned API*

You will setup an API to serve the result of the computed stats from the DB to a user on request:

1. Word Count: User can send the name/id of a document and is served the word count information in return
2. Noun / Verb Count: User can send the name/id of a document and is served the Noun / Verb count
3. If the user does not specify a particular document, he is served the statistics for all the documents
4. If the user specifies an ID/Keyword (such as ALL) for documents then the stats are combined for all documents and returned (Word : "Where", Count : 12841, Document : ALL, means the count was computed across all three documents) This should work for all computed statistics
5. Sentence Difference ***(Bonus)***: User can request top 10 most unique / dissimilar sentences in each document. This means that a request to the API with a specified document ID will return the top sentences which are most different than all other sentences. Apart from this, user can input a sentence and in response gets the most similar as well as the most different sentence, compared to his, from each document.

Evaluation:

Please Note that you are the designer of the application. This means that other than whats specified, you are free to design your APIs / use appropriate libraries for each task or you could simply write your own implementation. You are also free to decide which metrics and computational strategies are best for any task (Example: what is the metric for sentence difference and how to compute it). You will of course be evaluated for how well your methodology produces the desired results. Your code will be evaluated not only on how many tasks you complete but more importantly on how well does your application perform algorithmically and functionally. Therefore, **please make sure to comment your code / provide additional documentation explaining your choices and the reason you think they are the right way to do the job.**

Submission:

You can submit your assignment in one of the following two ways:

1. Commit your assignment on your Github account and share the repository
2. Bundle your submission as a Zip/RaR file and send it via email

Please feel free to contact us if you have any queries.