This report uses the files from Digital Humanities Project and the English dictionary (at https://github.com/dwyl/englishwords) along with the SnowballStemmer to identify all the non-English words through a mapper function. This mapper function gives an output as a string of non-English words that are stored in the intermediate space which is further used as input for the reducer function. The reducer function then takes each of the words in the identified non-English string and then gives the total count- thus achieving the goal of this project.

The following parts of the assignment have been achieved:

a) Well documented code with a Map-reduce function provided as "RKant_Final.ipynb".

b) All the functions including the mapper and reducer have been defined and documented in detail with their input and outputs.

   Output from the mapper and reducer have been printed at those steps where they have been executed and can be accessed in the attachment "RKant_Final.ipynb" or attachment "RKant_Final.pdf".

c) My Project Submission folder can be accessed using the below url :
   https://buffalo.app.box.com/folder/151531497073

d) The mapper and reducer output from my program has also been provided separately as "Output from the Mapper_Reducer.pdf". This contains the mapper and reducer output for each of the songs provided as part of this assignment.

e) My algorithmic approach and program design can further be optimized by:
   1) Using an object-oriented modelling design
   2) Using more intermittent variables or try catch procedure to handle any failures
   3) Using advanced NLP techniques and packages to identify language-based vector approach such that English-words are weeded out more efficiently.

f) This parallel programming method definitely speeded-up the computation time as my CPU has 4 cores and while 1 core was working with a given set of files the other cores were parallelly working on different set of files during both of the mappers and reducer procedures. As showcases in the time tracking stats, this led to huge processing time reduction where total mapper time was near to 10 seconds while the total reducer time was near to a small fraction of 1 sec ~ 0 seconds.

   This approach therefore definitely had an advantageous impact on the total processing time if compared to processing done on each of the files, sequentially.