Developer Guide:

Functions, which will process the data for the output of this application are most important part of this project.

Function:

1. Get\_Countries():

This function iterates through the data set and appends the “visitor\_country” value to a list named “countries”. Then we made a set named “countries\_set” to remove repetitions from the list. After this, we made a list called “countries\_weights” for storing the number of repetitions of each country.

These pieces of data are arguments to another function called “show\_countries\_hist()” which is responsible for setting up the histogram.

A screen shot of a computer program

Description automatically generated

1. Get\_Browsers():

This function starts another iteration in the data set and collects the “visitor\_useragent” value of each log and inserts it to a list called “browsers”. During this iteration, it also slices the string from beginning to the index of first “/”. This is because we wanted to remove the rest of the string and only have name of the browser in the list. After this, we created a set called “browsers\_set” and removed the repeated items from the list. after this, we made another list called “browsers\_weights” to store the number of repetitions for each country.

These pieces of data are arguments to another function called “show\_browsers\_hist()” which is responsible for setting up the histogram.

A screen shot of a computer program

Description automatically generated

1. user\_timeSpent():

This function also iterates through the data set and collects “event\_readtime” value of users. For this function we used a dictionary to prevent the extra loops. Each user that have spent some time reading documents, will have a key and value (equal to sum of reading times) in the “users\_time” dictionary.

At the end, we have used this dictionary to return top 10 reader of our data set.

A screen shot of a computer program

Description automatically generated

1. Also\_likes():

This functionality is consisted of 3 functions, “also\_like\_user()”, “also\_likes\_doc()”, and “also\_likes()”. Also\_likes\_user gets a user uuid and returns the whole documents that the user have read. Also\_likes\_doc get a document uuid and returns all the users who have read this document. Also\_likes function uses these two functions to work. It takes a document uuid and (optionally) a user uuid and return a list of other documents as suggestion. These documents are coming from other users who have read the document and also read other documents.

A screen shot of a computer program

Description automatically generated

Reflections on programming language and implementation:

Python libraries like “tkinter” and “matplotlib” made the implementation process easier for us. Using these libraries made us needless to other languages, we could estimate our code execution time, directly translate .json data set to python objects and also show the excluded data on the GUI.

About limitations, we tried to run the coding for 3 millions lines sample but unfortunately visual studio code has crashed. I assume we could use other libraries to reduce the execution time and make the process easier for CPU. Although the pythonic code is much faster than other iterations in other languages, it was still impossible to CPU to handle 3 million lines of data. Pandas could be the best option for us to standardise the data sets and handle the input.