# Crawler Code Review

Reviewed By: Jibreel Natsheh – Computer Engineering Student/ Summer Intern at ASAL Technologies

Code is written by: Aseel Arafeh

You can find the code In the following Github repo:
https://github.com/AseelArafeh/searchEngine/tree/master/mySearchEngine

Introduction:

This review will be divided as my mentor asked to done the review into the following sections:

1. Take your colleague's crawler code and database, and do the necessary changes to make it run on your machine:

   I have inserted the database into Mysql Workbench on my computer and changed the code's database configurations to suit my computer's configurations.

2. Perform a comprehensive code review for your colleague's code, consider the following aspects:
   a. The overall design and architecture:
      The code design is well structured and each file contains only the necessary code but there is some lack of files that could make the code runs better and make it more scalable.
   b. Code modularity:
      The database file ("db.py") is a good database manager that have helped a lot in performing the SQL queries but for the database tables there should have been files for it – file for each table to control the operations of each table (deal with the table as Class and the table's instance as an Object).
   c. OOP standards:
      There was lack of methods for the classes and some functions with the database had no functions to operate from the code like determine which website to crawl, where I was supposed to go to the Mysql Workbench each time to operate a lot of functions manually like adding links to be scrapped.
   d. Reusability:
      The code is not efficient to be reusable as each time the code runs it keeps tracking all links in the database which takes a lot of time especially when using the code for many days.
   e. Scalability:
      To make the code more scalable there should have class for each table contains all relations with other tables and methods to operate the needed functions.
   f. Good/bad practices found:
      There is many good practices like using good functions and using comments to declare what each function is made for, and for the bad practices it mostly was in the database as in the hyperlinks table the id was not set to auto increment and for the foreign keys it should be cascade and not restrict on delete and update.
   g. Any possible bugs in the flow:

Not all crawled links are valid and some was valid but also is logically wrong and leads to go into infinite loop while running the program.

3. Run your colleague's crawler to get data from the following websites: aljazeera.com, msn.com, python.org
4. While running the application in step #3, include the following in your report:
   a. Total run time per website:
      As the crawler was going forever while running on any website it was impossible to calculate executing time but for the average time for indexing a page in each website is as following:
      Aljazera: 12.0 sec
      MSN: 2.93 sec
      Pyhton: 3.72
      Note that as long as the code runs the average time/ page increases due to that it keeps looping on the links table continuously.
   b. Average CPU & memory usage per website:
      a. Aljazera:
         a. CPU:8.65%
         b. RAM: 4.821 GB
      b. MSN:
         a. CPU:6.63%
         b. RAM: 4.264 GB
      c. Python:
         a. CPU:16.23%
         b. RAM: 4.539 GB
   c. Total number of database records the crawler was able to find per website:
      It was hard to determine this thing but in general after running the code for like 24 hours it crawled over than 1M links.
   d. Total number of valid/invalid records per website, and the accuracy percentage:
      The number of invalid records was growing exponentially while the valid increases linearly.
   e. Record any crashes/exceptions if encountered:
      In Python.org it shows a problem when the content contained non ASCII not UTF-8 characters like symbols and emoji.
5. Run the crawler to get data from the same 3 websites above, but run it in an infinite mode for 10 hours. (i.e. crawl the first website, then the second, then the third, then the first … infinitely for 10 hours), and record the following:
   a. The CPU usage over time, for example take the CPU usage reading every 15 minutes ( or less if you like): 8.547%
   b. The Memory usage over time, record the readings same as the previous point: 4.524 GB
   c. The database size over time (To calculate the database size, refer to THIS ARTICLE):
      Unable to get the overall size as I had to delete all instance before each time running the code to get better estimation.
   d. Draw three charts to plot your readings in the previous points (the X-axis for the chart is the time, and the Y-axis is the CPU/Memory/DB usage):
   e. Write a brief conclusion about your findings:
      In conclusion the indexer is good and well-designed but for the links scrapper it is not working well and also for the algorithm that moves between links it is slow and could be faster and also there should have been more focus on the OOP concepts and models.
6. Submit your report in PDF.