# COMP 4047 Internet and World Wide Web

## Group Project: Design and Implementation of a Search Engine

## 1. Project Specification

In this project, you will design and implement a search engine. This search engine gathers information from the web and then serves users' requests for searching web contents. In particular, it does not collect information from the web pages listed in a given blacklist (e.g., these web pages contain offensive contents), and it does not record the keywords listed in another given blacklist (e.g., these words are rude or offensive).

### 1.1 Gathering Information

Write a Java program to gather words in English from HTML documents and their corresponding URLs.

*Blacklists:* The following two blacklists are given in text file format:

- Blacklist of words: This blacklist contains the words that the search engine would not record and serve (e.g., these words are rude or offensive).
- Blacklist of URLs: This blacklist contains the URLs of the web pages that the search engine would not collect information from (e.g., these web pages contain sensitive or offensive contents). These URLs are expressed in one of the following two forms:
  1. Regular form (e.g., http://www.abc.org/index.html).
  2. Compact form using the wildcard character "*" (e.g., http://www.abc.org/weapon/* represents all the URLs under the folder named "weapon" in the file hierarchy such as http://www.abc.org/weapon/list.html and http://www.abc.org/weapon/gun/ak37/index.html),

*Data structures:*

1. URL Pool and Processed URL Pool are used to store URLs, where URL Pool can store at most $X$ URLs (where $X$ is a design parameter; see Step 1 of the following algorithm).
2. Design suitable tables to efficiently store the words and their corresponding URLs.

*Algorithm:*

1. Input and initialization: The user is prompted to provide: i) a seed URL which serves as a starting point for web search, and ii) the values of the parameters $X$ and $Y$ (where $Y$ is used in Step 4). Assign this URL to URL Pool and set Processed URL Pool to empty.
2. Retrieve and remove a URL from URL Pool on a first-come-first-served basis, add this URL to Processed URL Pool, and get the corresponding web page.

3. Process the web page obtained in Step 2 as follows:
   3.1 Extract all words from this web page. For each word, if it is not listed in the given blacklist, store the following items: i) the word, and ii) the URL and the title of the web page.
   3.2 Extract the file name (excluding the path and file extension) in `src` attribute and all words in `alt` attribute of `img` tag from the HTML document. For each word, if it is not listed in the given blacklist, store the following items: i) the word, ii) the URL of the web page, and iii) the image URL specified in the `src` attribute of the image tag.
   3.3 Extract all URLs from this web page. For each of these URLs, add it to the `URL Pool` if it satisfies four conditions: i) it is not listed in the given blacklist, ii) it does not appear in `URL Pool`, iii) it does not appear in the `Processed URL Pool`, and iv) the number of URLs in the `URL Pool` is less than *X*.
4. If the number of URLs in the `Processed URL Pool` is less than *Y*, go to Step 2; otherwise, stop.

## 1.2 Serving Requests

Write a Java program to serve users' requests and support the following:

1. **Keyword Matching:** The user's query contains a keyword where this keyword can be any word. The program finds the URLs of the web pages which contain the given keyword.
2. **Multiple-keywords Matching:** The user's query contains multiple keywords and the logical relationship among these words can only be "AND", "OR" or "NOT". A spaces " " between two keywords is equivalent to an "AND" operation. The word "OR" between two keywords is equivalent to an "OR" operation. A minus sign "-" just before a word is equivalent to a "NOT" operation.
3. **Keyword Matching for image:** User can specify to search for images by keywords (either single keyword or multiple-keywords). The program finds the URLs of the web pages that contain images whose file names or `alt` attributes match the keywords.

The program composes a web page to list the fulfilled URLs with their title. If the user specify to search for images, the fulfilled images will also be displayed in the resulting web page. Then the web server sends this page to the user.

Use **Spring** (https://spring.io/guides/gs/serving-web-content/), which is a Java web server and framework, for your implementation.

## 2. Project Information and Announcement

The project information and announcement (e.g., grouping, schedule for demonstration, etc.) are posted on moodle of the course.

## 3. Assessment Criteria

1. You must use Java to implement the specification given in Section 1.

2. You are NOT allowed to use any third-party libraries (except the Spring framework mentioned in Section 1) nor database management systems (e.g., MySQL, Oracle, etc.).

3. *Design:* There are many alternatives to design a search engine. Your design will be assessed in several aspects:
   (i) Performance: Is it fast? Is it storage-efficient?
   (ii) Program structure: Is it easy to understand, maintain and modify your programs?

4. *Implementation:* Your implementation will be assessed in three aspects: i) Does it implement all the specified functions? ii) Are there bugs? iii) Is the implementation efficient?

5. *Documentation:* Your source programs should contain clear and useful comments. Your report should clearly contain sufficient details.

6. *Marking scheme:*
   - **Mid-point assessment (10%)**
   - **Final assessment (90%):** The marking scheme is specified in the participation forms. In addition, cooperation is very important and all group members **must evenly share the work**. This is one of the assessment criteria. If your group members could not share the work evenly for any reasons (e.g., a member is ill for a long period), please inform Dr. Tony K. C. Chan as soon as possible or there may be mark deduction.

## 4. Submission and Demonstration

### 1. Forming Groups

**Each group has three students**. Form your own group and email the names and student IDs of your group members to Mr. Liao Xuankun at xkliao@comp.hkbu.edu.hk **on or before 12:00 noon, 24 September 2021 (Friday)**. If we do not receive your group information by this deadline, we will form a group for you. If your group has only two students, we may assign a student to your group or dismiss your group; if your group has more than three students, we may remove one or more students from your group or dismiss your group. The finalized grouping will be posted on moodle by **25 September 2021 (Saturday)**.

2. **Mid-Point Submission:** Prepare a mid-point report which: i) describes the design of the search engine (e.g., the data structures used to gather the collected information, the modules of the search engine and the functions of each module, etc.), ii) describes how the group members plan to share the implementation work (e.g., the modules to be implemented by each group member), and iii) includes a signed *Planned Participation Form* (which is available on moodle of the course). Submit this mid-point report by **12:00 noon, 11 October 2021 (Monday)** to Moodle. Mr. Liao Xuankun will arrange to meet and provide feedback to every group within the period 12−15 October 2021.

3. **Final Submission**

   **3.1 Report:** Each group prepares one report which fulfills the following requirements: i) it describes the details of the design and implementation of your search engine, ii) it is a Microsoft Word file, iii) its file name is ***group_y_report.docx*** where *y* is your group number, and iv) it contains the group number and the names & student IDs of all group members at the beginning. Each group submits one report by **12:00 noon on 8 November 2021 (Monday)** as follows:

   - Login into HKBU Moodle.
   - Select **COMP4047 Project Report**
   - Upload the file

   Email submission is NOT accepted.

   **3.2 Participation Form:** Submit a signed *Participation Form* (which is available on moodle of the course) to Dr. Tony K. C. Chan's mailbox by **12:00 noon on 8 November 2021 (Monday)**.

   **3.3 Program and Data Files:** Prepare the following files:

   - Program files (source files, executable files, etc.).
   - Data files which are obtained by executing your search engine with $X$=10, $Y$=100 and the following seed URL:

     https://www.comp.hkbu.edu.hk

     The data files contain the gathered words and their corresponding URLs.

   Each group (say, group *y*) packs the above files into one file named ***group_y.zip*** and submit this file by **12:00 noon on 8 November 2021 (Monday)** as follows:

   - Login into HKBU Moodle
   - Select **COMP4047 Program and Data Files**
   - Upload the file

   Email submission is NOT accepted.

4. **Demonstration and Assessment:** Demonstrate your search engine and explain its source code during **9 – 12 November 2021**. **Mr. Liao Xuankun** will arrange a suitable time slot for this demonstration session and inform you. Please note the following:

   - If you do not attend this demonstration session, you will be given zero mark for this project.
   - If you cannot explain your source code in the demonstration, it will be regarded as "suspected plagiarism" and it will be reported to the Department.
   - During demonstration, we will assess your search engine via a new seed URL, parameters *X* & *Y*, blacklist and ignore files.
   - During demonstration, you will be requested to compile your program from your submitted source files.