

Project 3: **Search Index**  
Search Engine Development  
Due: Sun, May 4 by midnight  
100 points

You are to create a search engine that searches the inverted index from the previous assignment. To accomplish this, you will create a single servlet called **SearchIndex.java**. Some requirements:

1. The inverted index should be created from the web pages in the cache directory when the servlet is first loaded (in the init method). The cache directory is located at Tomcat\webapps\ROOT\cache.
2. For simplicity, all queries will be assumed to use boolean ANDs. For example, the query *tall girls* is should generate a boolean search for *tall AND girls*. Only the top 10 results will be shown.
3. Before examining the inverted index, the query should be stemmed using the Porter stemming algorithm.
4. You will need to modify your InvertedIndex.java file from the previous project to search the inverted index. One method you should create is

```
public Vector<Integer> docsContainingQuery(String query)
```

This function should be called by SearchIndex with the given query, and the top 10 doc IDs matching the query should be returned. The doc IDs should be sorted according to the following scoring function:

$$\text{score} = 0.5 * \text{body TF} + 0.5 * \text{metatag score} + 0.9 * \text{title score}$$

where body TF is the number of times the word appears in the body of the document, metatag score is 1 if the term is present in the metatags (0 otherwise), and title score is 1 if the term is present in the title (0 otherwise).

5. To perform the ANDs, you should implement the intersect routines shown in the first chapter of the IR text and discussed in class.
6. The total number of results should be show on the SERP. Each result should be numbered, displaying the title of the document (Untitled if one isn't present) that links to the live web page, a snippet which is the first 100 characters of the body, the URL of where the page is located, and a cached link taking the user to the cached page. Ideally we would want to make bold the portions of the title and snippet that match the query, but this is not necessary.

**McChallenge Extra Credit:** You can receive 1% extra on your final grade in this class by implementing phrase search. The user should enclose their query in quotes "like this" when searching for a phrase, and only documents that contain the phrase should be returned. You will need to use the positions of the terms that are stored in the inverted index to accomplish this task.

Submit your completed **SearchIndex.java** and **InvertedIndex.java** files to Easel before it is due. (The file must be named correctly.)