CSC326 - Programming Languages

Lab 2 - Development Phase II

Frontend and Backend

Group Number:	1
Group Members:	Qiaopeng Yang (995264574) Wenzhong Zhang (996711278)
Lab Section	Monday 2-5PM
Report Date	October 27, 2013

Project Description

The search engine project in lab 2 is composed by the frontend and the backend.

Frondend is updated with CSS and JavaScript, making it more readable and dynamic. In the code, query to database is made when user queries a keyword. The return page is displayed after user hit “Submit” button. The result is formatted in a table with border = 1. The result page still keeps a count of words entered, but only first word is queried.

The backend is not visible to user. Tests have to be done to determine whether the python methods are written correctly. The backend implements a number dict() type data structures to simplify during programming time. PageRank Scores will be computed by the PageRank algorithm provided. Lexicon, Inverted Index, Form_to_list, PageRank, Doc Index tables those five tables have been stored in the .db database by sqlite. This sqlite persistent storage acts like a bridge between frontend and backend. The backend will generate the sqlite database in current working folder. The frontend will retrieve the data from the sqlite database and display it on the webpage.

Describe the design decisions made for the Frontend:

In the frontend, comparing to previous lab, the following feature is added:

- CSS, making website pleasant for viewing
- JavaScript placeholder for AJAX scrolling
- function queryDB, which takes a query word
- return of queryDB is put in Table

Describe the design decisions made for the Backend:

1. lexicon_table format: (wordID int NOT NULL PRIMARY KEY, word text NOT NULL)
2. inverted_index_table format: (wordID int NOT NULL PRIMARY KEY, doc_id text NOT NULL)
3. from_to_list format: (count int NOT NULL PRIMARY KEY, fromID int NOT NULL, toID int NOT NULL)
4. documentInfo1_table format: (count int NOT NULL PRIMARY KEY, doc_ID int NOT NULL, url text NOT NULL)
5. rank_table format: (DID int NOT NULL PRIMARY KEY, ranking text NOT NULL)
6. the above 5 tables are stored in dbFile.db

Describe the contribution for each member.

	Wenzhong Zhang	**Qiaopeng Yang**
Frontend	Added CSS, primarily background, button style, and logo frame	
	Made bottle reload itself on source file change by setting reloader=True	
	Rework on set() for search	
	Added error 403, 404 handling	
Backend	Code review	Researched and implemented PageRand algorithm
	Cleaned up debugging comments	Generated and stored Lexicon, Inverted Index, PageRank, Doc Index tables in persistent storage by sqlite in python
	Added test script dbex.py for querying the database for pagerank and urls	Added test script wordID.py to ensure database is functioning correctly
		Peer Code Review

Instructions to run your code.

Front end (Please run crawler.py first):

1. Open Shell, change current working directory to where HelloWorld.py is located
2. Run `python HelloWorld.py`
3. On the same computer, open a browser, enter http://localhost:8080/.
4. In the “search” field, type in a keyword and click on “submit” button.
5. The result page is displayed.

Back end:

1. Create a text file named “urls.txt” in the same folder as the crawler.py file, copy and paste some links the file. Here are some examples:
   http://individual.utoronto.ca/zhwzh308/
   http://www.torontocpr.com/?gclid=CJGZ_LKZuLoCFepFMgodlAoAmw
   http://www.eecg.toronto.edu/~ashvin/courses/ece344/current/
2. Open Shell, change current working directory to where crawler.py is located
3. Type `python crawler.py`
4. The crawler will print a message when it is returned.