**Project Organization:**

The program is designed in three different parts responsible for respective functionalities.

**1. Class LemurBuilder:** This class is responsible for pre-processing for lemur search interface. This class first reads all html documents. Parses the documents and builds inverted index in file. Files generated by this pre-processor are:

1. file1.txt : Contains

collection\_term ctf dtf offset length

ctf – collection term frequency

dtf – document term frequency

2. statfile.txt

Total Documents in collection

Unique terms in collection

Total terms in collection

Average doc length

3. file2.txt : actual index file.

4. file3.txt: File containing information of individual documents, terms in document and their frequency.

**2. LemurInterface:** This class accepts input requests. Performs the search in index file and returns the result.

**3. LemurProject**: Code reads the query file. Processes the queries and sends queries for search. Returns result.

**Compiling:**

Project is divided in two folders:

1. lemurBuilder:

Compile using

javac Stemmer.java LemurBuilder.java

Run using

java LemurBuilder

java

2. lemurproject:

Copy generated files: file1.txt, file2.txt, file3.txt, statsfile.txt and docid.txt to this folder. (It already has default ones.)

Compile using:

javac Stemmer.java LemurInterface.java rawParser.java LemurProject.java

Class rawParser is used to parse raw query file.

Class xmlParser is also present to parse XML query file if needed. Currently only rawParser is used in project (as suggested by professor)

Porter stemmer is used for the project.

Run Using

java LemurProject number resultfilename.txt

1st arg: number - question number from 1 to 5

2nd arg: name of file where you want to store results