1) It is located inside the src folder in the Document, Postings, InvertedIndex, and IndexMain.

2) First, the JSON file was read using the library JSON-simple for java. The read-in objects were then converted to a document object which stores all its important information. From there the buildIndex algorithm was used. It mapped each of the terms to a list of postings that contained the document name. Number and positions. It also created 2 maps that mapped the sceneid to the length of the text and the playid to the length of text. That created the inverted index. Then I created multiple methods to serve as the API all allowing a user to gather different information on the list. Some of the methods are getTermBy() which get all the plays or scene(depending on what the user ask for) which contain the term, along with getPhrase which does the same but with phrases as opposed to terms. The methods allow a user to easily call them to allow many different kinds of quires to be searched as all that needs to be changed is what method is called and what the query is. In addition, the methods when given a query like the one in term0.txt will treat or as all the document with thee more than you and all the document with thou more than you instead of all document with thee + thou more than you.

3) Libraries used:
java.util.ArrayList -- Used to store data
java.util.Collections -- Called to sort data
java.util.HashMap -- Used to store data
java.util.HashSet -- Used to store data
java.util.Map -- used to store data
java.util.Set -- used to store data
java.io.File -- used to write and read data to file
java.io.FileReader used to write data to file
java.io.FileWriter -- used to write data to file
java.io.IOException -- used to handle errors
java.io.PrintWriter - used to write data to file
org.json.simple.JSONArray -- Used to read in JSON file
org.json.simple.JSONObject -- Used to read in JSON file
org.json.simple.parser.JSONParser -- Used to read in JSON file
org.json.simple.parser.ParseException -- Used to read in JSON file

4) The count may be misleading in comparing different scenes because counts by themselves tell you only if the words are in the scene nothing else about the words. The word might be mentioned many times but be in a different context or it could only be mentioned once by name but talked about for the rest of the scene. In both cases, the count misleads you on the relevance of the scene. To fix this you need more information like the location of itself and what rems are nearby.

5) The queries that took the longest were the first 2 this is because it had to get the list for all individual parts of the query and then it had to combine it into a singular set getting rid of duplicate terms.

6) Average Scene Length: 1200 words
Shortest Scene: antony_and_cleopatra:2.8: 47
Longest Scene: loves_labors_lost:4.1: 7988
Shortest Play: comedy_of_errors: 81504
Longest Play: hamlet: 165129