**User Manual:**

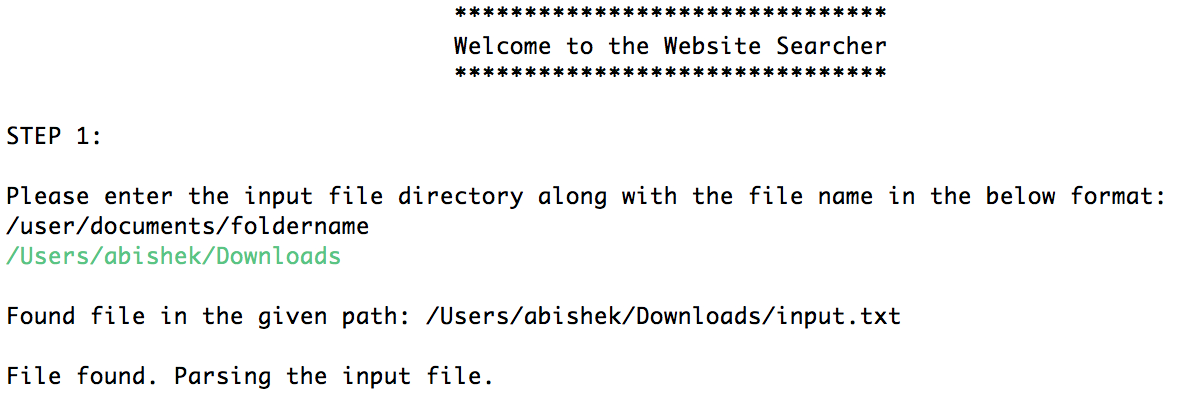
**STEP1:**

**Input File:**

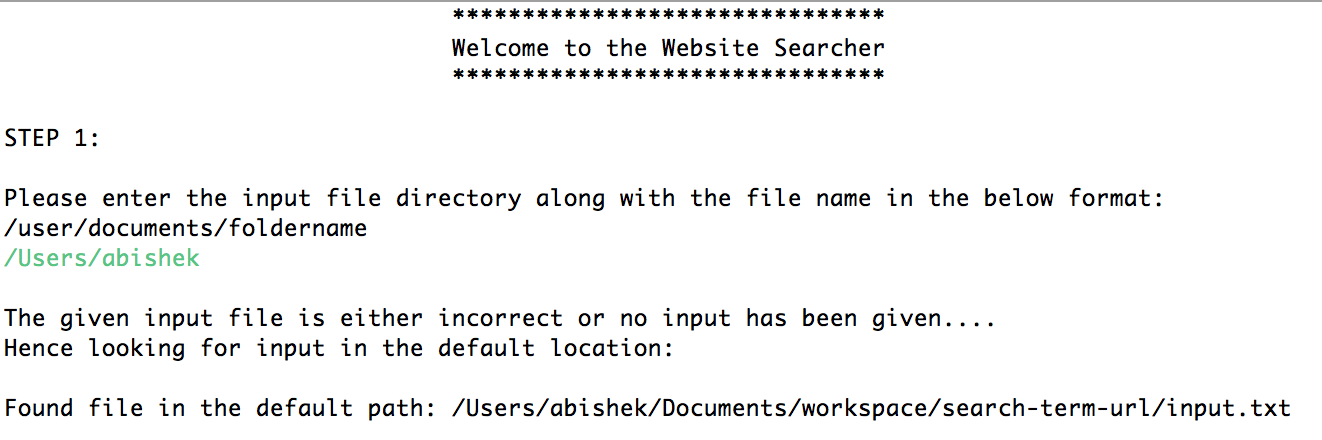
* Input file name should be “input.txt”.
* Please have the “input.txt” file in the default location (classpath directory of the project).
* User can specify the location of the input file in the format mentioned below with an example.
* /user/documents/foldername

**Validation:**

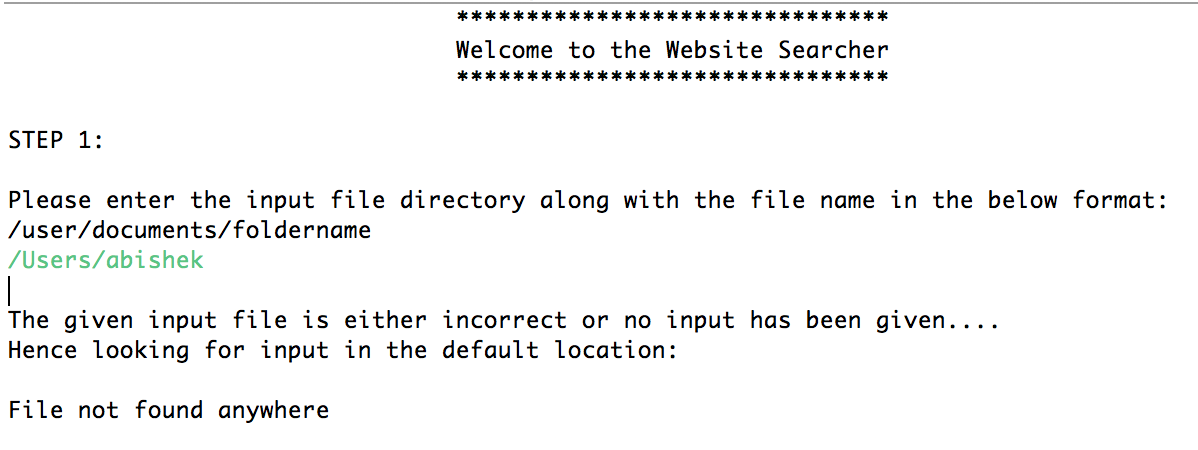
* If the user specified location is a directory and the input file exists, the input file will be parsed and the list of URLs’ will be extracted to begin the search.



* If the input file is not available in the mentioned directory or an incorrect location is specified by the user, the program will look for the “input.txt” from the default location. The default location of the file is the classpath directory of the project.



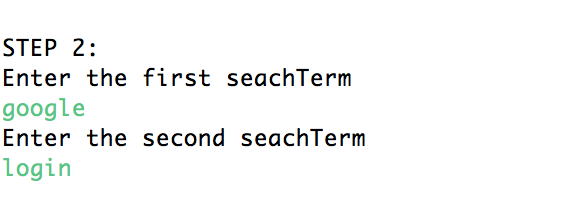
* If the file is not available in the user specified or the default location, then the program exits with the message “File not found anywhere”.



**STEP2:**

**Search Terms:**

* User enters SearchTerm1 and SearchTerm2 (Cannot be empty).
* The search terms can be two different words. The program handles the logic of searching either of the URL’s for a match on the page.



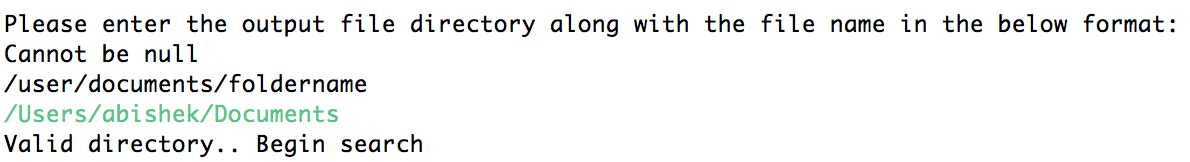
**STEP3:**

**Output File:**

* The data will be stored in the output file “results.txt”.
* User can specify the location of the output file in the format mentioned below with an example. Cannot be empty.
* /user/documents/foldername

**Validation:**

* If the user specified location is a directory and the “results.txt” file does not exist, the program creates Threads to begin searching for the search terms in the URL’s.



* If the user specified location is a directory and the “results.txt” file already exists in the location, the file will be deleted before the program begins execution.

