* ~~1. Search page: Create a main web page with a central search box for users to enter a query and a button to execute the search.~~
* ~~2. Search action: When the button is clicked in the UI, the search text must be passed to this engine’s search function.~~
* 3. Search function in the engine: Provide in your search engine a general “search” function that receives a string as parameter. Add any other parameter you consider helpful for the implementation.
* 4. The search algorithms: The search function in turn must call the previously defined

algorithms:

* + a. Now that you have a real use case, optimize your algorithms for the goal of retrieving the best results, faster, cleaner, and that better suit the user’s information needs.
  + b. Structure your code in a way that can be used in the web application.
  + c. The documents corpus will be the already provided Twitter entries.
* 5. The results page: create a web page that displays the list of documents found for the query and in the calculated order/ranking.
  + a. Each result record must represent a document from the corpus so it must have at least the following properties:
    - i. Title (in tweets we can use a substring with some initial characters of the full tweet text)
    - ii. Summary description (probably the whole tweet text)
    - iii. Creation date/time
    - iv. URL (used to link to document details page)
    - v. Other items you consider relevant for the results page.
* 6. The document details page: a page to display the whole document's information.
  + a. Display other relevant properties present in the corpus for each document.