

---

## Part 4: User Interface and Web Analytics

---

### User Interface

It is now the time to give your search engine a user interface (UI) and apply some web analytics on it. You are asked to provide a Web application for entering the search query, displaying the search results and usage statistics.

Use the simple Python web framework **Flask** that runs its own development web server (a project boilerplate will be created in class as a hands-on part in the Web Analytics seminar).

Project content:

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.

## Web Analytics

You are required to provide a robust mechanism for effective tracking and analyzing how people use your website and the search engine's use.

As this is for educational purposes you can store data in memory (the use of a database for long time persistence is optional and if used it must be very well documented in order for teachers to reproduce in their environments for evaluation).

1. **Data collection:** You can for example collect and store the following actions:
  1. For HTTP
    - Requests data
    - Clicks
    - HTTP Sessions data
  2. For queries: number of terms, order, etc.
  3. For results (documents)
    - clicks on documents.
    - to what query where related.
    - ranking of clicked documents.
    - dwell time: the time between clicks on a result document and coming back to the results page.
  4. For user context (visitor):
    - browser, OS/computer/mobile, time of the day, date
    - *IP address, country, city (optional bonus point)*
  5. *For sessions (optional bonus point)*
    - *time based sessions: queries of a user in the same sit down (physical session).*
    - *missions: sequence of queries with the same goal (logical session), there can be multiple missions within a physical session.*
    - *research missions: span across multiple sit downs, days, weeks, etc.*
  6. All other information that you find interesting (justify in your report)
2. **Data storage:** Design a data model to store the information that you collect about the website usage that you collect from above point. Consider a star schema for example with the following tables:
  - a. Session
  - b. Click
  - c. Request
3. **The analytics dashboard:** Create a web page that displays the usage statistics.
  - a. Define key indicators, metrics, and reports.
  - b. Show how people use your website and search engine.
  - c. Provide graphs with explanations of what is displayed.