Assignment 1 – Boolean Retrieval with Pyserini
Due Date: 26 August 2025 (midnight)

Objective

The goal of this assignment is to introduce you to the fundamentals of **Boolean Retrieval** and the process of building a simple **search engine index** using the **Pyserini** library (a Python interface to Lucene/Anserini).

By completing this assignment, you will:

- Understand the concept of the **inverted index** and its role in retrieval.
- Learn how to **index documents** using Pyserini.
- Perform **Boolean retrieval** queries (`AND`, `OR`, `NOT`) on a small corpus.
- Apply **light preprocessing** (lowercasing, stopword removal, stemming).

Dataset

You are provided with a small corpus of **15 short documents (sentences)**. Each document has an **ID** (`d1` … `d15`) and a **sentence of text**.

Example documents:

- d1: *"The cat chased a small mouse into the garden."*
- d2: *"A friendly dog played fetch by the river."*
- d3: *"BM25 is a ranking function widely used in search engines."*
- … (complete list will be given in the assignment file)

Tasks

Step 1 – Preprocessing

Before indexing, perform **light preprocessing** on the documents:

- Convert text to **lowercase**.
- Remove **stopwords** (common words like *the, is, and, of*).
- Apply **stemming** (use Porter stemmer provided in Pyserini).
- Remove punctuation and extra whitespace.

(Hint: Pyserini provides `StandardAnalyzer` and supports stemming at indexing time.)

Step 2 – Indexing the Corpus

1. Store the 15 documents in a **JSONL file** (each line = JSON object with `"id"` and `"contents"`).
   Example:

```
{"id": "d1", "contents": "the cat chased a small mouse into the garden"}
{"id": "d2", "contents": "a friendly dog played fetch by the river"}
```

2. Use **Pyserini's indexer** (`pyserini.index.lucene`) to create a Lucene index.
    - Use the **JsonCollection** format.
    - Enable stemming (`--stemmer porter`).
    - Store positions, docvectors, and raw text for later retrieval.

Step 3 – Boolean Retrieval

Use **Pyserini's** `LuceneSearcher` to perform Boolean queries on your index.

- Test queries with **logical operators**:
    - `dog AND cat`
    - `dog OR cat`
    - `dog AND NOT cat`
    - `(bm25 OR tf-idf) AND retrieval`
- Display the **list of matching document IDs** for each query.
  (Ignore scores – treat results as binary relevant/not relevant.)

Step 4 – Reporting Results

Each group must submit a **short report** (3–4 pages, PDF) that includes:

1. **Preprocessing explanation –** Describe the steps you applied to clean the documents.
2. **Indexing process –** Show how you indexed the documents using Pyserini (with code snippets).
3. **Boolean queries –** Provide at least **5 Boolean queries**, the matching documents, and a short explanation of why those results are correct.

Deliverables

1. **Code files** (notebook).
2. **Report (PDF)** containing description, code snippets, queries, and results.

Notes

- This is a **group assignment**: exactly 3 students per group.
- You are encouraged to use **Google Colab** or your own environment.
- You may consult Pyserini documentation and tutorials, but your code and report must be original.