# Search engine for articles related to COVID-19

## Purpose

The search engine is created with the purpose of finding articles related to COVID-19 that are of interest to a user depending on the keywords they will enter in the search engine. Through this search engine, a user will be able to expand their knowledge of how COVID-19 affects various areas of our daily lives.

## Functionality

A user will be able to search for articles of interest from a collection of over 500 articles related to COVID-19. The user will be able to enter keywords into the search engine, navigate through the results that will appear, sort the results, as well as view the articles that catch their interest.

## Text analysis and index construction

The index construction process is as follows:

### Data collection

The articles that will be used for this search engine's document collection come from the following location:
https://www.kaggle.com/jannalipenkova/covid19-public-media-dataset.
This collection contains over 350.000 articles related to COVID-19.
These articles examine the impact of COVID-19 on areas such as economics, technology, business development, psychology and many others.

### Article preprocessing

From the above collection, 550 articles were randomly selected, which are in csv format and have the following fields: *author*, *date*, *domain*, *title*, *url*, *content* and *topic_area*.
In the contents of the articles (content field) there were characters that needed to be removed, such as: € and □, but also some that needed to be replaced by others, such as ™ which was replaced by the apostrophe '.
The preprocessing of the original csv file was done using a program written in Python. The encoding of the file is UTF-8.
Afterwards, each article in the collection will be converted into a *Document* (Build Document).
Each *Document* will then be analyzed using the *StandardAnalyzer* provided by Lucene (Analyze Document).

### Construction of the index

For the construction of the index, the document unit will be the *article* and the fields that will be used are the *title* (title) and the *content* (content). Each *Document* will be added to the index via the *IndexWriter*.

## Search

To implement the *search* function, the *Search User Interface* is used which provides the user with the ability to enter the keywords of their choice in a search box.
The program then parses the user's keywords using Lucene's *QueryParser*. The keywords will then be analyzed with the same *Analyzer* that analyzed the *Documents*, and then a *query* will be created.
Using the *query*, the *IndexSearcher* will search the index, in order to find the correct *documents* and then returns *TopDocs*, which contain the ranked results.

## Presentation of the results

The answer to the user's question will be displayed in a new window.
The user will be able to sort the results, as well as select an article of their interest.
When the user selects an article, it will appear in the current window, with the keywords the user used highlighted.