Data Engineering Course Project

Under the supervision of Dr. Saptarshi Pyne

INTRODUCTION

- **Objective**

  Develop a Video Search Engine Application with a comprehensive GUI.

- **Key Features**

  Integrated use of MongoDB, Neo4j, and MySQL databases for a comprehensive video search engine. Utilizing MongoDB for efficient video file indexing, Neo4j for managing video relationships, and MySQL for storing relational information, including crucial click-through data.

BACKGROUND RESEARCH

YOUTUBE TAGS:

- Tags are descriptive keywords you can add to your video to help viewers find your content. Your video’s title and description are more important pieces of metadata for your video’s discovery. These main pieces of info help viewers decide which videos to watch.
- Tags can be useful if the content of your video is commonly misspelled. Otherwise, tags play a minimal role in your video's discovery.
- Sometimes, the video creators are penalized when they add misleading tags so tags are very useful in searching.

https://support.google.com/youtube/answer/146402?hl=en

WORKFLOW

KIVY :

- Kivy, an open-source Python framework, stands at the forefront of our video search engine application's GUI development. Recognized for its cross-platform compatibility, Kivy empowers us to craft interactive and visually engaging user interfaces seamlessly. Its versatility and ease of use make it an ideal choice for creating the Search Query Panel (SQP), Search Result Panel (SRP), and other components. Leveraging Kivy's capabilities, our goal is to deliver a user-friendly interface that enhances the overall experience of searching and interacting with video content. With Kivy, we aim to strike the perfect balance between functionality and aesthetics in our video search engine application.

BACKEND

BACKEND

MySQL

Storing click-through information, utilizing SQL querying capabilities to maintain structured data integrity

MongoDB

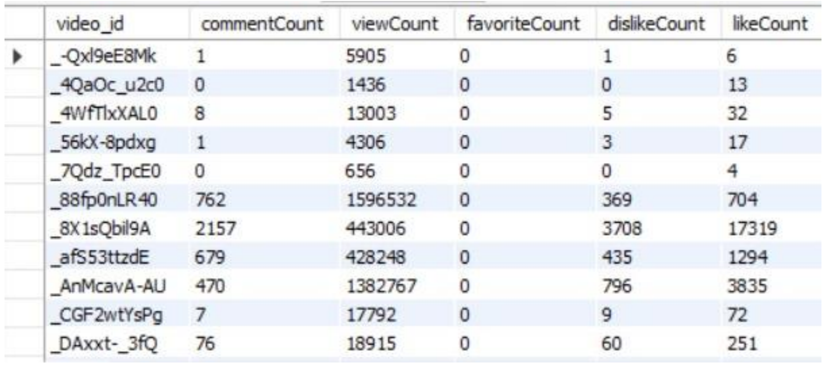
Ideal for storing and indexing video files with flexible schemas and efficient binary data handling.

Neo4j

Managing intricate relationships within video datasets, offering intuitive representations and efficient traversal

WORKING WITH SQL

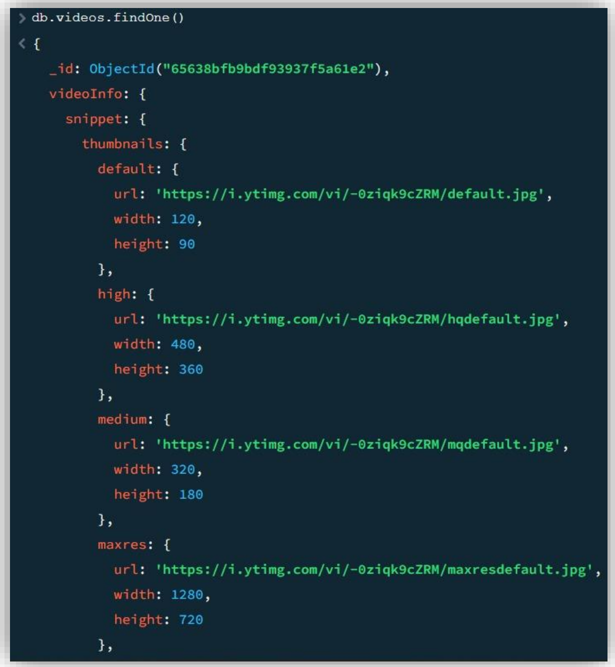
The VideoDatabase class, utilizing the MySQL connector, manages video-related data in the search engine application. It features methods for creating structured tables for video statistics and engagement analytics. The class integrates external data from JSON files into the MySQL database. With 'performing_search' and 'performing_search_2' methods, it enables searches based on video ID, extracting relevant information on video statistics and engagement analytics. Importantly, this class exemplifies the integration of MySQL, MongoDB (for indexing), and Neo4j (for relationship management), presenting a holistic approach to handling diverse facets of video data within the comprehensive search engine application.



video_id	commentCount	viewCount	favoriteCount	dislikeCount	likeCount
_-Qxl9eE8Mk	1	5905	0	1	6
_4QaOc_u2c0	0	1436	0	0	13
_4WfTlxXAL0	8	13003	0	5	32
_56kX-8pdxg	1	4306	0	3	17
_7Qdz_TpcE0	0	656	0	0	4
_88fp0nLR40	762	1596532	0	369	704
_8X1sQbil9A	2157	443006	0	3708	17319
_afS53ttzdE	679	428248	0	435	1294
_AnMcavA-AU	470	1382767	0	796	3835
_CGF2wtYsPg	7	17792	0	9	72
_DAxxt-_3fQ	76	18915	0	60	251

WORKING WITH MongoDB

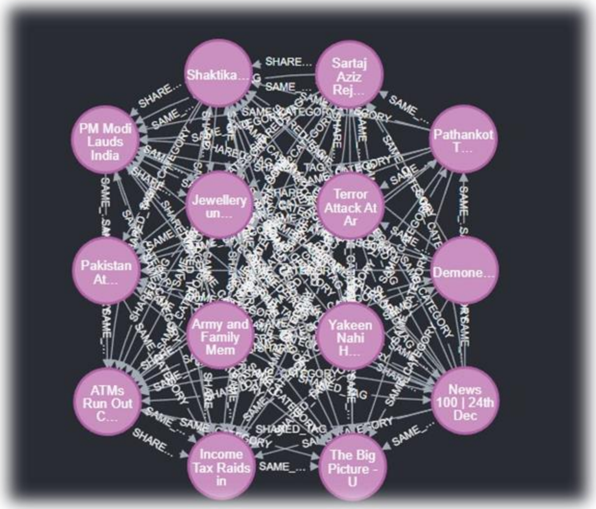
We initialize a MongoDB connection with flexibility using pymongo's MongoClient. It establishes a 'videos' collection with the create_collection method, dedicated to storing organized video data. The insert_videos method iterates through specified JSON files, excluding 'statistics' from videoInfo for separate handling, ensuring a tidy MongoDB collection. The perform_search method employs MongoDB's $or query operator for flexible searches in video titles, descriptions, and tags. The get_info method retrieves detailed video information based on ID from the 'videos' collection. Together, these functions showcase efficient data management and retrieval within MongoDB for video-related content.



```
> db.videos.findOne()
< {
    _id: ObjectId("65638bfb9bdf93937f5a61e2"),
    videoInfo: {
      snippet: {
        thumbnails: {
          default: {
            url: 'https://i.ytimg.com/vi/-0ziqk9cZRM/default.jpg',
            width: 120,
            height: 90
          },
          high: {
            url: 'https://i.ytimg.com/vi/-0ziqk9cZRM/hqdefault.jpg',
            width: 480,
            height: 360
          },
          medium: {
            url: 'https://i.ytimg.com/vi/-0ziqk9cZRM/mqdefault.jpg',
            width: 320,
            height: 180
          },
          maxres: {
            url: 'https://i.ytimg.com/vi/-0ziqk9cZRM/maxresdefault.jpg',
            width: 1280,
            height: 720
          },
```

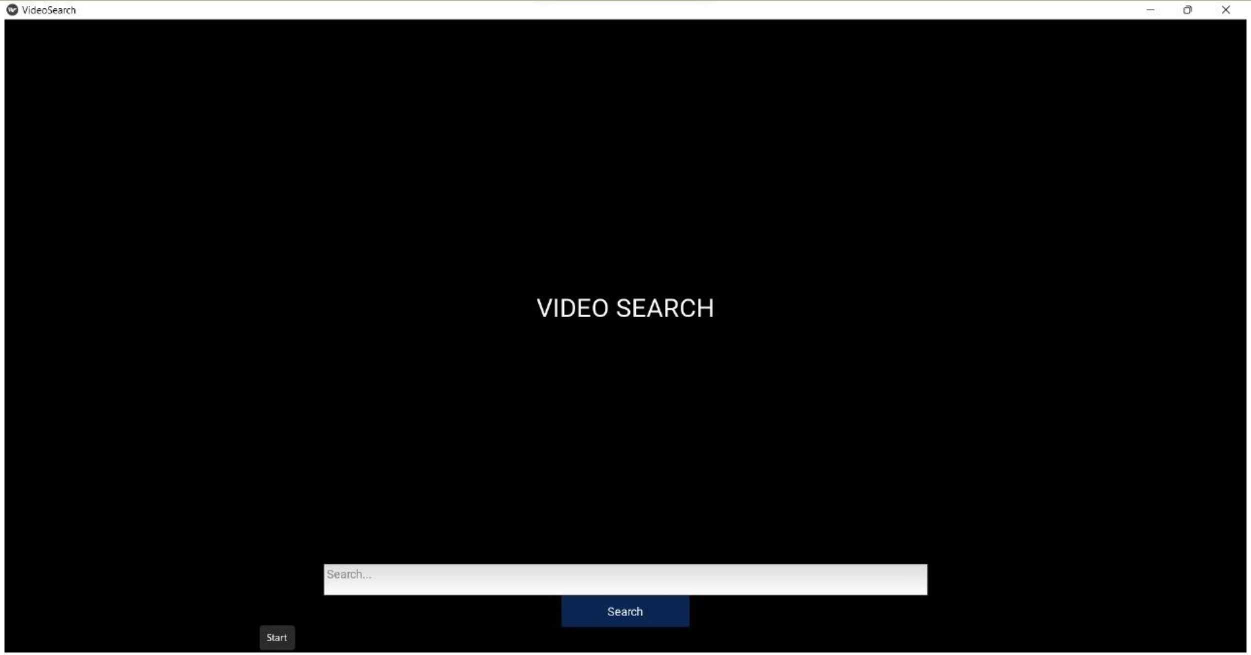
WORKING WITH Neo4j

We introduce a VideoGraphDatabase class to interact with a Neo4j graph database, using parameters like URI, username, and password for connection. The connect method attempts connection, displaying success or error messages. Data insertion involves creating a mega JSON file, a powerset of given data, manually merged into Neo4j. The get_most_connected_videos method executes a Cypher query to identify the two most connected videos based on 'SHARED_TAG' relationships. Error handling addresses connection exceptions. In the main block, a VideoGraphDatabase object is instantiated, connection details are provided, and the connect method is called for testing accessibility to the Neo4j database.



FRONTEND

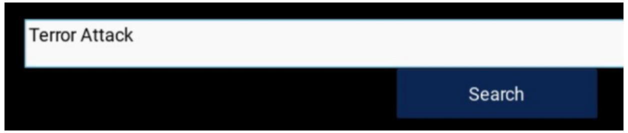
HOME PAGE

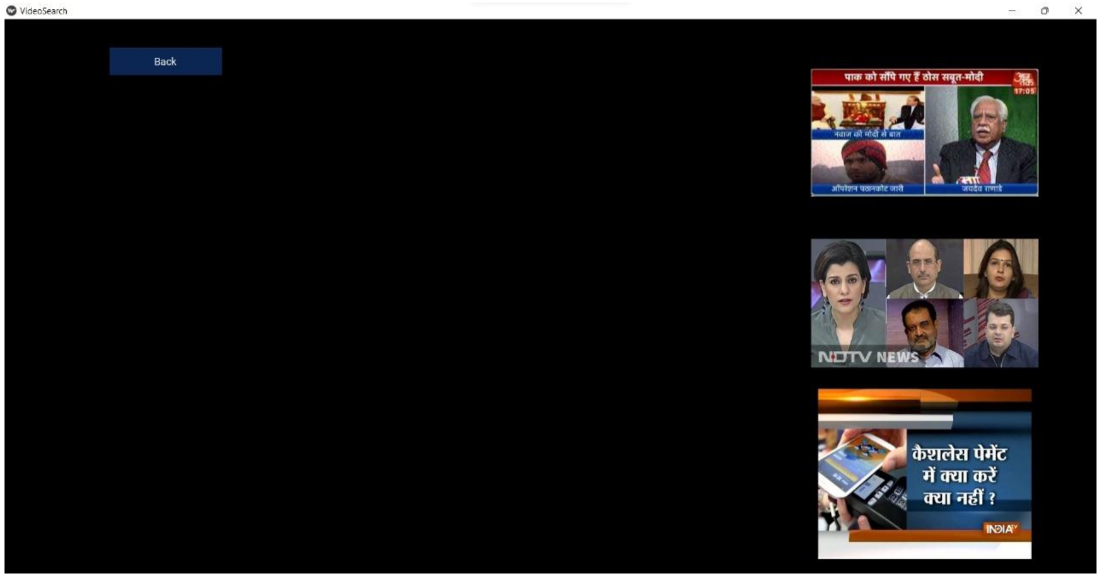


Here we start searching with the key words

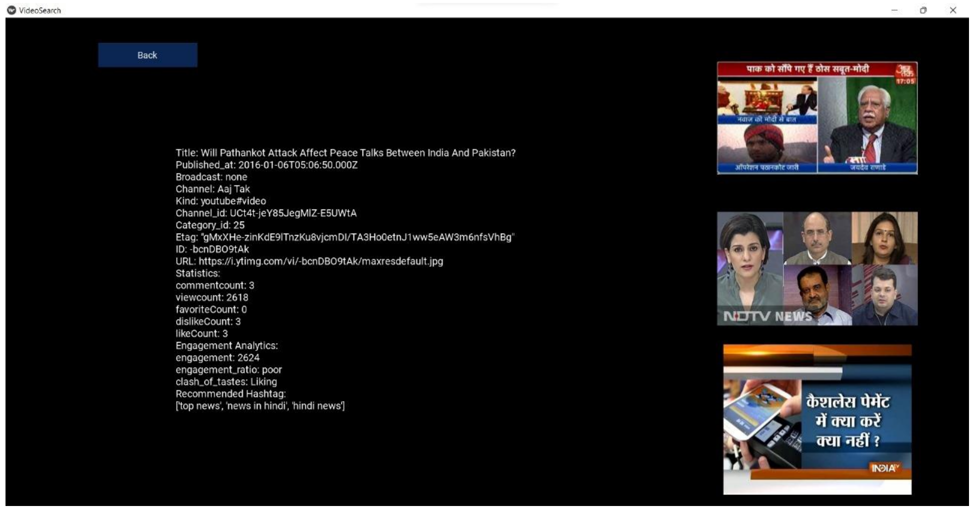
SEARCH RESULTS

Suppose,
We search for Terror Attack

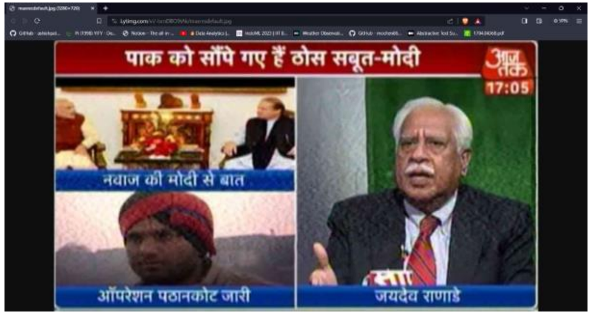




Click Through Content


VideoSearch
Back
पाक को सौंपे गए हैं ठोस सबूत-मोदी
NDTV NEWS
कैशलेस पेमेंट में क्या करें क्या नहीं ?

```
Title: Will Pathankot Attack Affect Peace Talks Between India And Pakistan?
Published_at: 2016-01-06T05:06:50.000Z
Broadcast: none
Channel: Aaj Tak
Kind: youtube#video
Channel_id: UCt4t-jeY85JegMlZ-E5UWtA
Category_id: 25
Etag: "gMxXHe-zinKdE9lTnzKu8vjcmDI/TA3Ho0etnJ1ww5eAW3m6nfsVhBg"
ID: -bcnDBO9tAk
URL: https://i.ytimg.com/vi/-bcnDBO9tAk/maxresdefault.jpg
Statistics:
commentcount: 3
viewcount: 2618
favoriteCount: 0
dislikeCount: 3
likeCount: 3
Engagement Analytics:
engagement: 2624
engagement_ratio: poor
clash_of_tastes: Liking
Recommended Hashtag:
['top news', 'news in hindi', 'hindi news']
```

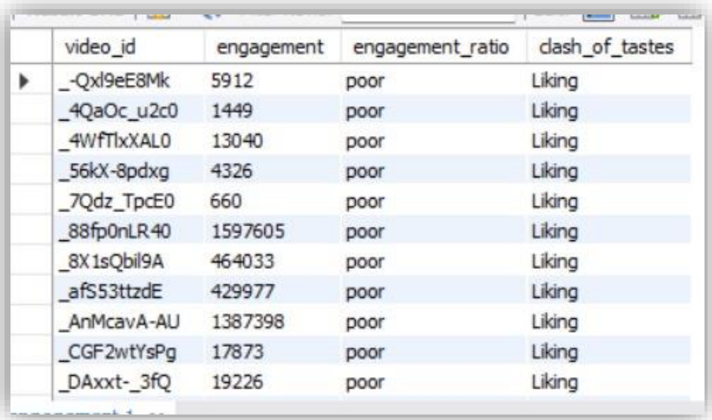


The thumbnail of video clicked gets popped up in browser

ADDITIONAL FEATURES

ENGAGEMENT TABLE:

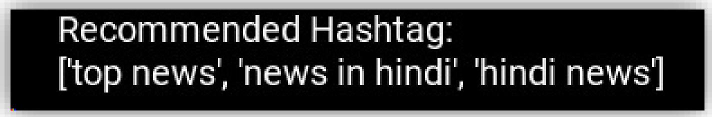
- We built an analytics table for the better understanding of the statistics.
- We created columns engagement, engagement ratio and clash of tastes.



video_id	engagement	engagement_ratio	clash_of_tastes
_-Qxl9eE8Mk	5912	poor	Liking
_4QaOc_u2c0	1449	poor	Liking
_4WfTlxXAL0	13040	poor	Liking
_56kX-8pdxg	4326	poor	Liking
_7Qdz_TpcE0	660	poor	Liking
_88fp0nLR40	1597605	poor	Liking
_8X1sQbil9A	464033	poor	Liking
_afS53ttzdE	429977	poor	Liking
_AnMcavA-AU	1387398	poor	Liking
_CGF2wtYsPg	17873	poor	Liking
_DAxxt-_3fQ	19226	poor	Liking

HASHTAGS:

- We give users relevant hashtags for easy searching of videos.



```
Recommended Hashtag:
['top news', 'news in hindi', 'hindi news']
```

Thanks

Does anyone have any questions?

GROUP 13

Dhyan | Rahul | Kovidh | Manihas