



Twitter Search Application - Team 2

[Github Link](#)

**Course: 16:954:694:01 - Data Management for Advanced
Data Science Applications**

Team Members:

Sai Adarsh Kasula (sk2837)
Krit Shreeram Gupta (ksg124)
Himani Hooda (hh660)
Vanshika Ram Gurbani (vg460)

Department of Statistics & Data Science

Rutgers University

New Brunswick,NJ

Supervised by

Prof. Ajita John

Note: Presentation to Cohort is completed (Cohort -1)

1. Introduction

Twitter is a highly popular social media platform with millions of users posting tweets every day. The Twitter search application uses Twitter's extensive database to let users quickly find relevant information on any topic. Users can search for tweets by username, hashtag, or text snippet, with additional options to narrow down results. This project explores different data storage designs and retrieval schemas, the use of various database technologies, and how datasets are divided for efficient storage. It also includes developing a caching model to speed up data retrieval. The report will cover the dataset and storage, search application design, caching strategy, search queries used, and the results obtained.

2. Dataset & Exploratory Analysis (By Vanshika Ram Gurbani)

The dataset 'corona-out-3' is in JSON format and contains about 112k documents from April 2020, during the Covid period. A sentiment analysis revealed a slightly negative overall sentiment of 0.02. The dataset includes approximately 90K unique users, and the tweets fall into various categories shown in the figure below. Each document includes user and tweet information, stored separately for better organization. The top 10 languages used in the tweets are displayed in Figure 2.2. For further analysis and visualizations, check the GitHub repository's Data_Analysis.ipynb file.

@brithume	1496
@Quirinale	1465
@benwikler	987
@oxfara	846
@yalim_funda	734
@realDonaldTrump	700
@narendramodi	636
@aajtak	625
@CrazyinRussia	587
@IngrahamAngle	559

dtype: int64

Fig 2.1: 10 Most mentioned users

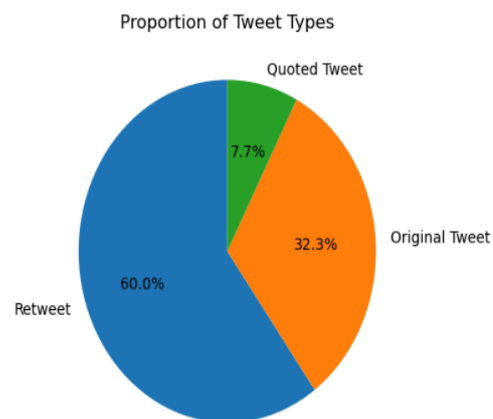


Fig 2.2: Proportion of Tweet type

3. System Architecture

System architecture is a crucial component that must be outlined, at least preliminarily, before beginning the development of any application. In our project, prioritizing this step was essential. The system architecture delineates the application's overall workflow, highlighting the major components and their interactions within the larger system. Considering the range of designs and functionalities possible with our dataset, we opted to focus on those features that would be most pertinent to an average Twitter user. Figure 3.1 below illustrates our final architecture and its key functionalities.

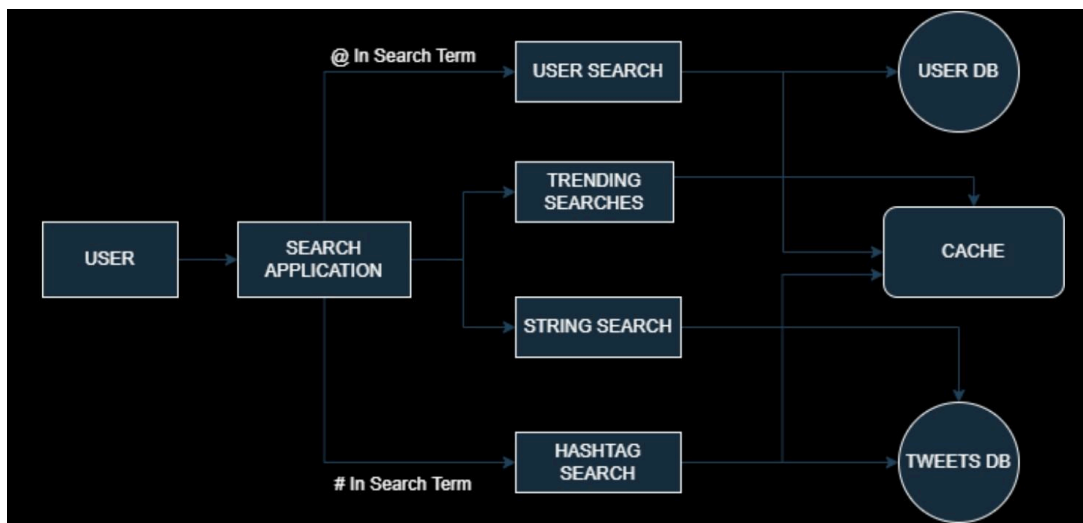


Fig 3.1 System Architecture

The Search Application's user interface, developed with FLASK- Python web framework—acts as the main channel for user interaction and navigation through various URLs tailored for different search types. The interface incorporates three primary functionalities: searching by User, Hashtag, and Search String/Term. Additionally, it offers users the ability to quickly view Trending Searches, including top Users, Hashtags, and Tweets. To enhance search efficiency, a caching system using a Python dictionary as the core data structure was implemented, further discussed in detail

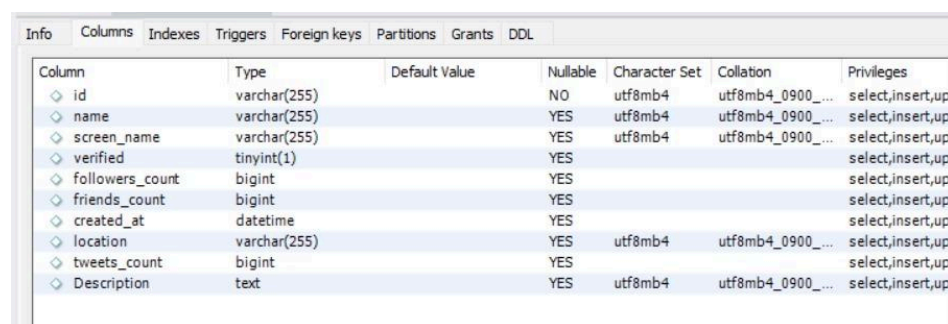
The files were processed to separate User information and tweet details. User Information was stored in a MySQL relational database. On the other hand, the core functionality of the Twitter application involves processing tweets, which requires extensive read and write operations. A non-relational database like MongoDB was chosen. A detailed examination of each of these components will be provided later in this report.

4. Data Storage (By Krit Sreeram Gupta)

As outlined in the previous section, User, and tweet data, are stored in two distinct database systems tailored to their specific needs and usage patterns.

4.1. User Data Storage

This section highlights the design strategy and how the user data was extracted and transferred using Python to a MySQL database. Prior to initiating the data extraction and loading process, essential libraries like "mysql.connector" and "json" were imported. The users table was created in the database, with fields such as ID, name, location, verification status, and others to store user information. The data extraction and loading process involved defining two key functions. The `loadData()` function reads the `corona-out-3` files, extracting each JSON object, which could be source tweets, retweets, or quoted tweets. For each tweet, the user details of the author were extracted and passed to the second function, `UserInsert()`. This function processed the user information by extracting the ID from the user object and checking if this user already existed in the database. If not, the user's details were inserted into the database.

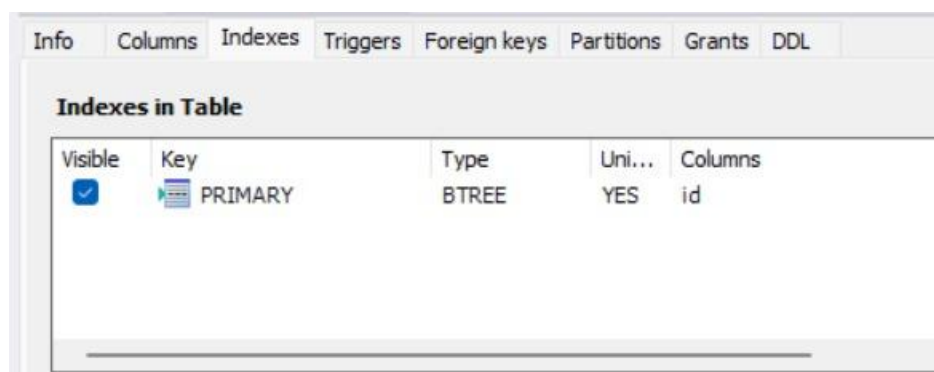


Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
id	varchar(255)		NO	utf8mb4	utf8mb4_0900_...	select,insert,up
name	varchar(255)		YES	utf8mb4	utf8mb4_0900_...	select,insert,up
screen_name	varchar(255)		YES	utf8mb4	utf8mb4_0900_...	select,insert,up
verified	tinyint(1)		YES			select,insert,up
followers_count	bigint		YES			select,insert,up
friends_count	bigint		YES			select,insert,up
created_at	datetime		YES			select,insert,up
location	varchar(255)		YES	utf8mb4	utf8mb4_0900_...	select,insert,up
tweets_count	bigint		YES			select,insert,up
Description	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,up

Fig 4.1.1 Attributes in the Users Table

Efficient handling of duplicates was crucial due to the high frequency of tweets, retweets, and quoted tweets by the same users. After extracting user information, a total of 90,000 unique users were identified, and eight different attributes for each were stored, enhancing the dataset's utility and richness for further analysis and querying.

After the data loading process, indexing was implemented on two specific columns to enhance query performance. MySQL automatically creates a primary index on the ID column, which serves as the primary key for user identification. This utilizes the Binary Tree type, which facilitates quicker search and retrieval operations.



Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	id

Fig 4.1.2: Indices in Users Table

4.2. Tweets Data Storage

MongoDB was chosen as a non-relational database. MongoDB simplifies the storage, search, and retrieval of tweet data, eliminating the need for a tweet parser. Below functions were then implemented to manage tweet data:

- ``insert_tweet()``: Inserting a tweet into the database.
- ``find_tweet()``: Checking if a tweet already existed in the database to prevent duplicate entries.
- ``retweet_update()``: Updating the retweet count whenever a retweet was encountered.
- ``process_tweets()``: Reading the input JSON files line by line, checking for duplicates, and inserting both tweets and retweets.

Approximately 112,000 unique tweets and retweets were processed and stored, as depicted in Figure 4.2.1. The data loading process ensured that every retweet was linked to its original source tweet, enabling any necessary drill-through features in the search application, as shown in Figure 4.2.2. After the data loading, two indexes were created on the fields "User_Id" and "Text", as in Figure.



```

_id: ObjectId('662914d94d73db0d9c499964')
created_at: "2020-04-25 07:30:12"
Tweet_Id: "1253949413191344128"
Text: "India's war with Corona is ongoing.

        Play your part and make sure no o..."
Hashtag: Array (empty)
User_Id: "207809313"
User_Name: "BJP"
Retweet_Count: 59
Likes_Count: 1870

```

Fig 4.2.1: MongoDB Summary & Fig 4.2.2: MongoDB Schema for tweets

Indices for Tweets_collection

Name and Definition	Type	Size	Usage	Properties
id _id ↑	REGULAR	2.6 MB	99 (since Wed Apr 24 2024)	UNIQUE
Text_text _fts (text) _ftsx ↑	TEXT	24.1 MB	16 (since Wed Apr 24 2024)	

Fig 4.2.3: Indices in MongoDB

5. Search Implementation (By Sai Adarsh Kasula)

Searches form the core part of the application, in this section, the implementation and working of the search functionalities is discussed.

5.1. User Search

The search feature helps locating users and their information in the application. When initiating a search, users type the "@" symbol followed by a search term it searches for other users. The application utilizes a search function that queries the database to retrieve and display user-related results. To improve performance, a caching strategy is employed. The cache is first checked for any existing matches. If matches are found, the results are immediately retrieved from the cache. If no matches exist, the function fetches the results directly from the database and stores them in the cache for future searches of the same query.

Furthermore, the search function retrieves and caches the three most recent tweets of each user from the MongoDB Tweets database.

Users can view the most recent tweets of selected authors, sorted by `followers_count` and `verified_status`. After the search results are displayed, users can select desired authors by entering a number from 1 to 5. The application then retrieves the corresponding tweets from the cache and presents them to the user.

5.2. String Search

The application's functionality includes an advanced search for specific content within tweets, utilizing a text index on the tweet's text field. To refine results, the system removes standalone stop words from the queries. If a search query does not start with '#' or '@', it is treated as a text search, displaying the five most recent tweets matching the criteria. Due to the varied nature of text searches, these results are not cached.

5.3. Hashtag Search

The application also features a dedicated search function for hashtags, where users can initiate a search by entering the "#" symbol. When a user conducts this search, the application retrieves and displays the top five hashtags. These results include exact matches and similar hashtags, ranked by the frequency of their appearance in the database. For a deeper exploration, the application offers a drill-through feature; users can select any of the displayed hashtags to view the top three most relevant tweets associated with them. These tweets are sorted by their recent creation date. This seamless process ensures that the tweets are ready to be displayed immediately.

5.4. Trending Searches

Trending Users:

The "Trending Users" feature of the application provides users the most popular users based on their tweet count and number of followers. A function is activated to retrieve the trending user data. To optimize performance and minimize database load, caching strategy is implemented. This approach ensures that users can quickly discover the most influential users on the platform.

Trending Tweets:

The application includes a "Trending Tweets" feature that displays the most popular tweets based on a composite score derived from retweets and likes. In calculating this score, a weight of 60% is given to retweets, while 40% is attributed to likes. The top 10 tweets are then selected based on this composite score. For each trending tweet, the application displays the username, retweet count, likes count, the date and time the tweet was created, the text of the tweet, and any associated hashtags. These details are presented in descending order of their composite score..

Trending Hashtags:

The "Trending Hashtags" feature analyzes all tweets and counts the frequency of occurrence for each hashtag. The hashtags are then ranked in descending order based on their frequency of appearance. The user interface displays the top 10 hashtags, accompanied by their respective total counts.

6. Caching (By Himani Hooda)

6.1 Cache Implementation

Caching is a technique employed to accelerate data access by storing frequently accessed data in a temporary storage location called a cache. This approach allows for quick retrieval of data when needed, instead of querying the database every time.

For this project, a Python dictionary was utilized as a cache to store user information and their tweets. A Cache class was implemented to provide efficient caching mechanisms, offering various methods for managing the cache. The class initialization method `__init__()` sets the maximum size of the cache to 15000 objects, approximately 10 MB of RAM. The eviction strategy employed is "least_accessed," which ensures that when the cache is full, the item with the lowest access count is removed first, retaining the frequently accessed items.

The cache is periodically saved to disk at a checkpoint interval of 300 seconds (5 minutes) to maintain resilience. The time-to-live (TTL) for cached items is set to None, allowing objects to remain in the cache until evicted due to capacity constraints or manual removal.

The `get()` method retrieves the value associated with a key from the cache and increments the key's access count. If the key is not present, None is returned. The `put()` method adds a key-value pair to the cache, evicting the least accessed item if the cache reaches its maximum capacity. The access count for the new key is initialized to zero.

6.2 Caching in Search Application

The caching mechanism played a crucial role in improving the search functionality for both username and hashtag searches. Before querying the database, the application checks if the required data is present in the cache. If found, the results are directly returned using the `get()` method. Otherwise, the database is queried, and the results are saved in the cache using the `put()` method for faster retrieval in the future.

The cache is stored in a checkpoint file at regular intervals using the `save_to_checkpoint()` method and can be restored using the `load_from_checkpoint()` method when the system restarts, ensuring resilience in case of sudden failures. Additionally, the cache checks the access counts field before putting items into the cache to remove items according to the eviction strategy.

By implementing caching mechanisms, the system's performance is enhanced, and the utilization of system resources is optimized, resulting in improved efficiency.

7. Search Application Design (By Sai Adarsh Kasula)

The user interface was developed using CSS and HTML, while the connection between the frontend and backend was made using the Python Flask library. Multiple HTML pages were organized into a separate template folder for streamlined management. Below figure 7.1 shows the home page of the twitter search application.

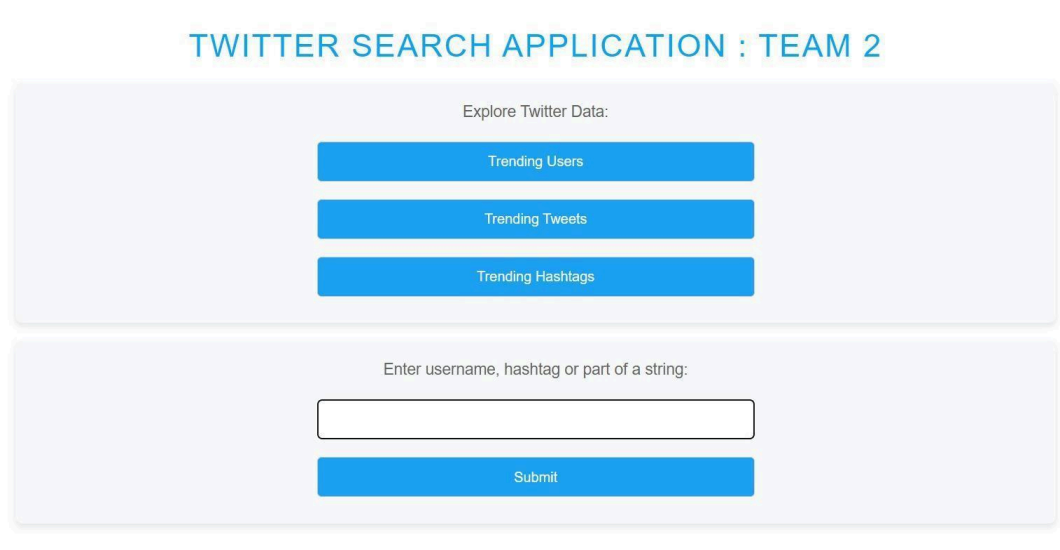


Figure 7.1: User interface home page

The home page of the UI features three buttons for trending users, tweets, and hashtags, and a search query input field and a submit button. These elements interact with the backend using

GET/POST methods, routing requests through functions in the `app.py` files. All HTML pages are stored in a "template" folder, accessed by Flask routing functions. Each query offers further drill-downs, such as displaying tweets linked to specific hashtags or usernames. Additionally, in the user results, hyperlinks in the Handle field connect to more detailed user profiles, as shown in the below figure 7.2.

Trending Users							
Name	Handle	Verified	Followers	Following	Location	Tweets	Bio
Barack Obama	BarackObama	Yes	116518121	607194	Washington, DC	0	Dad, husband, President, citizen.
Donald J. Trump	realDonaldTrump	Yes	78467254	46	Washington, DC	0	45th President of the United States of Americaus
CNN Breaking News	cnnbrk	Yes	57529057	120	Everywhere	0	Breaking news from CNN Digital. Now 56M strong. Check @cnn for all things CNN, breaking and more. Download the app for custom alerts: http://cnn.com/apps
Narendra Modi	narendramodi	Yes	55781248	2364	India	4	Prime Minister of India
Shakira	shakira	Yes	52250613	212	Barranquilla	0	👉 ME GUSTA Shakira & Anuel AA Nuevo Sencillo / New Single
CNN	CNN	Yes	47565193	1106	None	0	It's our job to #GoThere & tell the most difficult stories. Join us! For more breaking news updates follow @CNNBRK & Download our app http://cnn.com/apps
The New York Times	nytimes	Yes	46361159	904	New York City	1	News tips? Share them here: http://nyti.ms/2FVHq9v
BBC Breaking News	BBCBreaking	Yes	43014510	3	London, UK	0	Breaking news alerts and updates from the BBC. For news, features, analysis follow @BBCWorld (international) or @BBCNews (UK). Latest sport news @BBCSport.
Amitabh Bachchan	SrBachchan	Yes	41596464	1833	Mumbai, India	1	"तुमने हमें पूज पूज कर पथर कर डाला ; वे जो हमपर जुमले कसते हैं हमें झिंदा तो समझते हैं" ~ हरिवंश राय बच्चन
Salman Khan	BeingSalmanKhan	Yes	40094611	26	MUMBAI	0	Film actor, artist, painter, humanitarian

Fig 7.2: Username search having user handle as hyperlink

8. Results

8.1. User Searching: The results page displayed after searching for "@trump" includes all the essential fields on the UI. Beneath these results, there is a text box where users can enter a number from 1 to 5 to specify which author's tweets they wish to view.

User Search Results							
#	Name	Handle	Verified	Followers	Tweets	Description	Location
1	Donald J. Trump	realDonaldTrump	✓	76792576	0	45th President of the United States of Americaus	Washington, DC
2	Karti Q ★★★★★ -Text TRUMP to 88022	KarlukaP	✗	55107	0	That Cackling Conservative 🐔 sings too! #Trump2020-Music- https://store.cdbaby.com/artist/karlibonne https://music.apple.com/in/artist/karlibonne/29014046	None
3	MAGA-Joanne us Text TRUMP to 88022	JoanneTarpon07	✗	49662	1	#MAGA, TrumpTrain, #ZA, #KAG Christian Conservation, Florida Patriot, God+Country+Family, Nationalist #NRA #seanhamity, #loudobbs	United States
4	John Trumpfan	JohnTrumpFanKJV	✗	43817	3	Repentance toward God and Faith toward our Lord Jesus Christ. King James Bible Believer. Ye must be born again. Pro Life Pro Israel. God bless Pres Trump. #MAGA	Florida
5	TWITMO INMATE ★★★★★ Text TRUMP to 88022	TWITMO_INMATE	✗	33385	1	THE MORE YOU RESEARCH THE CRAZIER YOU SOUND TO IGNORANT PEOPLE... WWG1WGA FB @GenFlynn and Epstein didn't kill himself	United States

Enter the user number whose tweets you want to be displayed:

Submit

Fig 8.1.1: User Search Result

8.2. Hashtag Searching: Below is the results page after searching for the term “#trump”. Results are displayed by the number of tweets they appear in as shown below in Figure below.

Top 5 Hashtags Matching Your Search for "#trump"

Hashtag	Tweets Count
Trump	139
Corona	73
corona	43
NamasteTrump	40
HeilenwieTrump	22

Enter the hashtag related to which the relevant tweets you want to be displayed:

Submit

Fig 8.2.1: Top 5 Hashtags related to the term

From the results displayed, users can choose a specific hashtag by typing it into the search box below the results and clicking submit. For instance, if a user types in "Trump" as the hashtag, the UI will show the top three recent tweets featuring that hashtag, as illustrated in the figure below.

Tweets of #Trump

Username	Retweets	Likes	Created At	Text	Hashtags
Hatice Bingöl	0	0	2020-04-25 14:48:10	RT @hakana: ABD Başkanı #Trump 25 Şubatta sadece 15 #corona vakası olduğunu ve hepsinin iyileşeceğini ilan etmişti. Bugün ülkede nerdeyse 1...	['Trump', 'corona']
Zekiye Özyürek ⚡	0	0	2020-04-25 14:46:10	RT @hakana: ABD Başkanı #Trump 25 Şubatta sadece 15 #corona vakası olduğunu ve hepsinin iyileşeceğini ilan etmişti. Bugün ülkede nerdeyse 1...	['Trump', 'corona']
Ayşe	0	0	2020-04-25 14:45:56	RT @hakana: ABD Başkanı #Trump 25 Şubatta sadece 15 #corona vakası olduğunu ve hepsinin iyileşeceğini ilan etmişti. Bugün ülkede nerdeyse 1...	['Trump', 'corona']

Fig 8.2.2: Top 3 recent tweets of a hashtag

8.3. String Searching: The search results for the query "trump" display the top five tweets containing this string, shown on the UI.

Tweets relevant to "Trump"

User	Retweets	Likes	Created At	Text	Hashtags
Fitri Iskandar	0	0	2020-04-25 14:48:35	RT @Dandhy_Laksono: Jokowi minta ventilator dari Trump. Trump ngetwit siap bantu. Dosen ITB:	[]
snoesjoe	0	0	2020-04-25 14:48:33	MIRACLES. MALARIA DRUGS. MOTHER NATURE. TRUMP'S DANGEROUS CORONA 'CURES' https://t.co/9tRAy1epYa	[]
mary c	0	0	2020-04-25 14:48:31	RT @JoyceWhiteVance: What a coincidence that this happened AND that Trump blocked a highly regarded former DOJ IG as corona-IG. https://t.co/9tRAy1epYa	[]
Nikhil Kumar	0	0	2020-04-25 14:48:22	#IndiaWithPMModi God save India from corona and incapable leaders of India. Trump-Modi ki masti Gujratiyon per padi... https://t.co/8pT4yvWpHF	[#IndiaWithPMModi]
OurCountryNeedsEU	0	0	2020-04-25 14:48:22	RT @dennisdiclaudio: The elderly are being very unfair to Donald Trump.	[]

Figure 8.3: String search results

8.4. Trending Searches:

Trending Users: When the user clicks the Trending Users button on the Home Page, they are redirected to the page as shown in the Figure below, all the necessary information which are deemed necessary are fetched from the Users Database and displayed.

Trending Tweets: When the user clicks on the Trending Tweets button, they are redirected to the page shown below that displays the top 10 tweets based on the composite score which is discussed above.

Trending Tweets					
User	Retweets	Likes	Created At	Text	Hashtags
Quirinale	798	9524	2020-04-25 09:23:42	#25Aprile, il Presidente #Mattarella si è voluto recare all'Altare della Patria dove ha deposto una corona d'alloro... https://t.co/ev4b4DlqDf	[#25Aprile', 'Mattarella', 'Altare della Patria']
dr. Sheila Putri Sundawa	784	5498	2020-04-25 05:24:38	Ngomongin teori konspirasi corona sama orang yg percaya kalo bumi itu datar, I'm not a smart people, but please dud... https://t.co/aCiWbZg41m	[]
Quirinale	654	4062	2020-04-25 09:56:54	#25Aprile, nel 75° anniversario della #Liberazione il Presidente #Mattarella ha deposto una corona all'Altare della... https://t.co/vDeVcwEMQy	[#25Aprile', 'Liberazione', 'Mattarella']
Brit Hume	1474	1831	2020-04-25 11:27:12	And where is the evidence that Covid 19 is easily spread outdoors? https://t.co/YPcJXU1uqw	[]
euOnly In Russia	586	1352	2020-04-25 11:26:44	Corona disinfecting in Russia. https://t.co/AEF5ccDmM3	[]
• Sãç•	445	1532	2020-04-23 19:58:13	Quando eu ligo a televisão e fica falando só de corona vírus	[]
Ben Wikler	986	0	2020-04-25 12:51:03	Milwaukee's health commissioner has now tied 40 coronavirus infections to the April 7 election. https://t.co/0fGtSLKzTkm	[]
Funda	732	0	2020-04-25 12:43:26	Gözün çıkın corona 🤔 Ökece asabil öldük 🤔 muhtemel psikoloji ektedir 🤔 🤔 🤔 🤔 https://t.co/KoGSbVAMxZ	[#PideAlmayaDiyeÇikip]
Laura Ingraham	500	0	2020-04-25 13:53:37	A MUST READ... Coronavirus Restrictions: Government Bears the Burden of Proof Before Denying Freedoms National Rev... https://t.co/RcaAK9mwDs	[]
Gu Ru Thalaiva	422	0	2020-04-25 13:01:20	Thalapathy fans from Sivakasi Helped the Poor Family who are affected by this corona Crisis ! They have supplied th... https://t.co/ozoG6d9Ax8	[]

Fig 8.4.1: Trending Tweets

Trending Hashtags: Similar to the other trending searches, when the user clicks on the 'Trending Hashtags' button, they are redirected to the page as shown below which displays the top 10 most used hashtags across all the tweets and their counts.



Hashtag	Tweets Count
Corona	5133
corona	1636
Mattarella	1516
25Aprile	1476
Covid_19	1049
COVID19	877
AltaredellaPatria	806
PideAlmayaDiyeÇıkıp	777
coronavirus	707
Liberazione	700

Fig 8.4.3: Trending Hashtags

9. Conclusion

This project was a detailed exploration into data ingestion using multiple database technologies, specifically focusing on importing JSON data into MySQL and MongoDB databases. A search application was developed using Python Flask to demonstrate dynamic data retrieval based on user queries. Key enhancements such as caching with Python dictionaries and indexing significantly improved query performance, making the system more efficient. This endeavor not only provided a practical understanding of integrating diverse tools and technologies but also emphasized the critical role of thoughtful database design and optimization in boosting system performance.