In [ ]:
```
!pip install snscrape
```

In [ ]:
```
!pip install streamlit
```

In [ ]:
```
!pip install pymongo
```

In [ ]:
```
!pip install datetime
```

In [ ]:
```
! pip3 install git+https://github.com/JustAnotherArchivist/snscrape.git
```

In [8]:
```
%%writefile streamlit.py
import streamlit as st
import pandas as pd
import snscrape.modules.twitter as sntwitter
import pymongo
from datetime import date
import base64

st.title("Twitter Scraping")
# Set start and end dates
start_date = st.date_input("Start Date")
end_date = st.date_input("End Date")
# Creating a text box to enter the hashtag to search for,
hashtag = st.text_input("Enter hashtag to search for")

# To Check if the user entered a hashtag
if not hashtag:
    st.warning("Please enter a hashtag to search for.")
else:
# Twitter search query
    query = f"#{hashtag} since:{start_date} until:{end_date}"

# Creating a slider for selecting the number of tweets to scrape
    tweet_count = st.slider("Select number of tweets to scrape", min_value=10, max_

# Scraping Twitter data
    scraped_tweets = []
    for i, tweet in enumerate(sntwitter.TwitterSearchScraper(query).get_items()):
        if i >= tweet_count:
            break
        scraped_tweets.append({
            "date": tweet.date,
            "id": tweet.id,
            "url": tweet.url,
            "tweet_count": tweet.replyCount + tweet.retweetCount,
            "user": tweet.user.username,
            "reply_count": tweet.replyCount,
            "retweet_count": tweet.retweetCount,
            "language": tweet.lang,
            "source": tweet.sourceLabel,
            "like_count": tweet.likeCount,
            "hashtags": [hashtag for hashtag in tweet.hashtags]
        })

# To Display scraped data in a table
    if scraped_tweets:
        df = pd.DataFrame(scraped_tweets)
        st.write(df)
    else:
        st.warning("No tweets were scraped.")
```

```
# Create buttons to upload the data to MongoDB and download the data in CSV and JS
    if scraped_tweets:
# Connection to MongoDB
        client = pymongo.MongoClient("mongodb+srv://vijay:9944433644@cluster0.xezi
        db = client.scrap
        scrape = db.tweets
def main():
    if st.button("Upload data to MongoDB"):
# To Insert the scraped data into MongoDB
        scrape.insert_one ({
            'scrapped word': hashtag,
            'scrapped date': pd.Timestamp.now().strftime('%y-%m-%d'),
            'scrapped data': scraped_tweets
})
        st.success("Data uploaded to MongoDB.")
        st.markdown("""---""")

    if st.button("Download data as CSV"):
# To Download the scraped data as a CSV file
        csv = df.to_csv(index=False)
        b64 = base64.b64encode(csv.encode()).decode()
        href = f'<a href="data:file/csv;base64,{b64}" download="{hashtag}_tweets.c
        st.markdown(href, unsafe_allow_html=True)

    if st.button("Download data as JSON"):
# To Download the scraped data as a JSON file
        json = df.to_json(orient="records")
        b64 = base64.b64encode(json.encode()).decode()
        href = f'<a href="data:file/json;base64,{b64}" download="{hashtag}_tweets.
        st.markdown(href, unsafe_allow_html=True)

if __name__ == "__main__":
    main()
```

```
Overwriting streamlit.py
```

In [ ]:

```
#Run using below command
Write Streamlit code in jupyter Notebook.
Run the below commands in the command prompt,
jupyter nbconvert --to script Streamlit_Jupyter.ipynb
streamlit run app.py
```