

Reddit API'S for Fiance:


```
In [84]: import praw
import json
from datetime import datetime

# Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["finance"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert the created_utc timestamp to a human-readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date field
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance1_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to finance1_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: finance  
Fetched 101 posts. Data saved to finance1_API_Reddit_with_dates.json.
```



```
In [85]: import praw
import json
from datetime import datetime

# Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["Economics"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "POLICIES", "GDP", "TRUMP WIN"
]

# List to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Saving results to a JSON file
with open("finance2_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to finance2_API_Reddit_with_dates.json.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: Economics  
Fetched 228 posts. Data saved to finance2_API_Reddit_with_dates.json.json.
```



```
In [86]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["wallstreetbets"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "STOCKS", "U.S.A", "financial markets"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance3_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetched {len(posts)} posts. Data saved to finance3_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: wallstreetbets  
Fetched 250 posts. Data saved to finance3_API_Reddit_with_dates.json.
```



```
In [88]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["financialindependence"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "U.S. retirement plans", "U.S.A", "savings"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance4_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetched {len(posts)} posts. Data saved to finance4_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: financialindependence  
Fetched 249 posts. Data saved to finance4_API_Reddit_with_dates.json.
```



```
In [89]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["investing"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "after trump", "us market news", "U.S.A", "inflation", "economic crisis"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance5_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to finance5_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: investing  
Fetched 234 posts. Data saved to finance5_API_Reddit_with_dates.json.
```



```
In [90]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usa"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "finance", "economy after trump", "economy after trump"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance6_API_Reddit_with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to finance6_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: usa  
Fetched 205 posts. Data saved to finance6_API_Reddit_with_dates.json.
```



```
In [91]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usa"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "economy after trump"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert created_utc timestamp to a readable date
            created_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": created_date # Add the date to the post data
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("finance7_API_Reddit-with_dates.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetched {len(posts)} posts. Data saved to finance7_API_Reddit_with_dates.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: usa  
Fetched 205 posts. Data saved to finance7_API_Reddit_with_dates.json.
```


In [29]: **import** json

List of JSON file names

```
json_files = [  
    "finance1_API_Reddit_with_dates.json",  
    "finance2_API_Reddit_with_dates.json",  
    "finance3_API_Reddit_with_dates.json",  
    "finance4_API_Reddit_with_dates.json",  
    "finance5_API_Reddit_with_dates.json",  
    "finance6_API_Reddit_with_dates.json",  
    "finance7_API_Reddit-with_dates.json",  
    "finance11_API_with_dates.json",  
    "finance12_API_with_dates.json",  
    "finance13_API_with_dates.json",  
    'finance14_API_with_dates.json',  
    'finance15_API_with_dates.json',  
    'finance8_list_API.json',  
    'finance9_list_API.json',  
    'finance16_API.json',  
    'finance17_API.json',  
    'finance18_API.json',  
    'finance19_API.json'  
]
```

Initialize an empty list to hold all data

```
combined_data = []
```

Loop through each file and load the data

```
for file_name in json_files:
```

```
    try:
```

```
        with open(file_name, 'r') as f:
```

```
            data = json.load(f) # Load JSON data
```

```
            if isinstance(data, list): # Check if the data is a list
```

```
                combined_data.extend(data) # Add data to the combined list
```

```
            else:
```

```
                print(f"Warning: {file_name} does not contain a list. Skipping.")
```

```
    except FileNotFoundError:
```

```
        print(f"Error: {file_name} not found. Skipping.")
```

```
    except json.JSONDecodeError:
```

```
        print(f"Error: Could not decode JSON in {file_name}. Skipping.")
```

Save the combined data to a new JSON file

```
output_file = "combined_finance_API_Reddit_with_dates.json"
```

```
try:
```

```
    with open(output_file, 'w') as f:
```

```
        json.dump(combined_data, f, indent=4) # Save with indentation for readability
```



```
    print(f"Combined data saved to {output_file}. Total records: {len(combined_data)}")  
except Exception as e:  
    print(f"Error saving combined JSON file: {e}")
```

Combined data saved to combined_finance_API_Reddit_with_dates.json. Total records: 3482

Reddit API'S for Defence:


```
In [138]: import praw
import json
import pandas as pd

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["Military"]

# Keywords related to Trump, USA GDP, and the military
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "US military"
]

# Initialize list to store posts
posts = []

print("Fetching defence posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence1_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetchd {len(posts)} posts. Data saved to defence1_with_dates_API_Reddit.json.")
```

```
Fetching defence posts...  
Searching in subreddit: Military  
Fetched 248 posts. Data saved to defence1_with_dates_API_Reddit.json.
```



```
In [101]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["geopolitics"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence2_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to defence2_with_dates_API_Reddit.json.")
```

Fetching financial and economic posts...
Searching in subreddit: geopolitics
Fetched 248 posts. Data saved to defence2_with_dates_API_Reddit.json.


```

In [102]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["WarCollege"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america", "U.S. military doctrine", "future technologies"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence3_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to defence3_with_dates_API_Reddit.json.")

```

```
Fetching financial and economic posts...  
Searching in subreddit: WarCollege  
Fetched 219 posts. Data saved to defence3_with_dates_API_Reddit.json.
```



```
In [103]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usa"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america", "us army", "trump administration"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence4_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to defence4_with_dates_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: usa  
Fetched 236 posts. Data saved to defence4_with_dates_API_Reddit.json.
```



```
In [104]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["trump"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america", "soldiers"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence5_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to defence5_with_dates_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: trump  
Fetched 247 posts. Data saved to defence5_with_dates_API_Reddit.json.
```



```
In [126]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usarmy"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence6_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to defence6_with_dates_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: usarmy  
Fetched 117 posts. Data saved to defence6_with_dates_API_Reddit.json.
```



```
In [127]: import praw
import json
import pandas as pd # Import pandas for date handling

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["internationalnews"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us military", "america", "soldiers", "war"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "created_utc": submission.created_utc # Add timestamp
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Convert UNIX timestamp to readable dates
for post in posts:
    post["date"] = pd.to_datetime(post["created_utc"], unit='s').strftime('%Y-%m-%d')

# Save results to a JSON file
with open("defence7_with_dates_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetchd {len(posts)} posts. Data saved to defence7_with_dates_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: internationalnews  
Fetched 250 posts. Data saved to defence7_with_dates_API_Reddit.json.
```


In [35]: **import** json

List of JSON file names

```
json_files = [  
    "defence1_with_dates_API_Reddit.json",  
    "defence2_with_dates_API_Reddit.json",  
    "defence3_with_dates_API_Reddit.json",  
    "defence4_with_dates_API_Reddit.json",  
    "defence5_with_dates_API_Reddit.json",  
    "defence6_with_dates_API_Reddit.json",  
    "defence7_with_dates_API_Reddit.json",  
    "defence8_with_dates_API_Reddit.json",  
    "defence9_with_dates_API_Reddit.json",  
    "defence10_with_dates_API.json",  
    "defence11_with_dates_API.json",  
    "defence12_with_dates_API.json",  
    "defence13_with_dates_API.json"  
]
```

Initialize an empty list to hold all data

```
combined_data = []
```

Loop through each file and load the data

```
for file_name in json_files:  
    try:  
        with open(file_name, 'r') as f:  
            data = json.load(f) # Load JSON data  
            if isinstance(data, list): # Check if the data is a list  
                combined_data.extend(data) # Add data to the combined list  
            else:  
                print(f"Warning: {file_name} does not contain a list. Skipping.")  
    except FileNotFoundError:  
        print(f"Error: {file_name} not found. Skipping.")  
    except json.JSONDecodeError:  
        print(f"Error: Could not decode JSON in {file_name}. Skipping.")
```

Save the combined data to a new JSON file

```
output_file = "combined_defence_API_Reddit_with_dates.json"
```

```
try:  
    with open(output_file, 'w') as f:  
        json.dump(combined_data, f, indent=4) # Save with indentation for readability  
        print(f"Combined data saved to {output_file}. Total records: {len(combined_data)}")  
except Exception as e:  
    print(f"Error saving combined JSON file: {e}")
```


Combined data saved to combined_defence_API_Reddit_with_dates.json. Total records: 4043

Reddit API's for Healthcare


```

In [111]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["healthcare"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "us healthcare"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare1_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetched {len(posts)} posts. Data saved to healthcare1_API_Reddit.json.")

```

```
Fetching financial and economic posts...
Searching in subreddit: healthcare
Fetched 249 posts. Data saved to healthcare1_API_Reddit.json.
```



```
In [112]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["medicine"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "Public health policies"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare2_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to healthcare2_API_Reddit.json.")
```

Fetching financial and economic posts...
Searching in subreddit: medicine
Fetched 123 posts. Data saved to healthcare2_API_Reddit.json.


```
In [113]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usa"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "FDA approvals", "after trump healthcare", "pharmacy"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare3_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to healthcare3_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: usa  
Fetched 228 posts. Data saved to healthcare3_API_Reddit.json.
```



```
In [114]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["america"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "FDA approvals", "after trump healthcare", "pharmacy", "healthcare"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare4_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to healthcare4_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: america  
Fetched 238 posts. Data saved to healthcare4_API_Reddit.json.
```



```
In [115]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["america"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "medicine"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare5_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetches {len(posts)} posts. Data saved to healthcare5_API_Reddit.json.")
```

```
Fetching financial and economic posts...  
Searching in subreddit: america  
Fetched 244 posts. Data saved to healthcare5_API_Reddit.json.
```



```

In [9]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["unitedstatesofamerica"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "TRUMP", "USA", "TRUMP WIN", "healthcare"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare6_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetched {len(posts)} posts. Data saved to healthcare6_API_Reddit.json.")

```

```
Fetching financial and economic posts...
Searching in subreddit: unitedstatesofamerica
Fetched 244 posts. Data saved to healthcare6_API_Reddit.json.
```



```

In [8]: import praw
import json
from datetime import datetime

# Set up Reddit application credentials
reddit = praw.Reddit(
    client_id='uGkcVUHw7I45CAW4rEtWUw',
    client_secret='jr_mUYW7vAMTiZ8IqzKCYod-r3poHQ',
    user_agent='TrumpPolicyMarketAnalysis',
)

# List of subreddits for financial and economic news
subreddits = ["usa"]

# Keywords related to Trump finances, USA GDP, and stock market
keywords = [
    "trump again", "healthcare after trump"
]

# Initialize list to store posts
posts = []

print("Fetching financial and economic posts...")

for subreddit_name in subreddits:
    try:
        subreddit = reddit.subreddit(subreddit_name)
        print(f"Searching in subreddit: {subreddit_name}")
        for submission in subreddit.search(" OR ".join(keywords), sort="new", time_filter="all", limit=1000):
            # Convert UTC timestamp to a readable date format
            post_date = datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S')

            posts.append({
                "title": submission.title,
                "score": submission.score,
                "url": submission.url,
                "date": post_date # Add the date of the post
            })
    except Exception as e:
        print(f"Error with subreddit '{subreddit_name}': {e}")

# Save results to a JSON file
with open("healthcare7_API_Reddit.json", "w") as f:
    json.dump(posts, f, indent=4)

print(f"Fetchd {len(posts)} posts. Data saved to healthcare7_with_dates_API_Reddit.json.")

```

Fetching financial and economic posts...

Searching in subreddit: usa

Fetches 205 posts. Data saved to healthcare7_with_dates_API_Reddit.json.

In [65]: **import** json

```
# List of JSON file names
json_files = [
    "healthcare1_API_Reddit.json",
    "healthcare2_API_Reddit.json",
    "healthcare3_API_Reddit.json",
    "healthcare4_API_Reddit.json",
    "healthcare5_API_Reddit.json",
    "healthcare6_API_Reddit.json",
    "healthcare7_API_Reddit.json",
    "cleaned_healthcare8_list.json"

]

# Initialize an empty list to hold all data
combined_data = []

# Loop through each file and load the data
for file_name in json_files:
    try:
        with open(file_name, 'r') as f:
            data = json.load(f) # Load JSON data
            if isinstance(data, list): # Check if the data is a list
                combined_data.extend(data) # Add data to the combined list
            else:
                print(f"Warning: {file_name} does not contain a list. Skipping.")
    except FileNotFoundError:
        print(f"Error: {file_name} not found. Skipping.")
    except json.JSONDecodeError:
        print(f"Error: Could not decode JSON in {file_name}. Skipping.")

# Save the combined data to a new JSON file
output_file = "combined_healthcare_API_Reddit_with_dates.json"
try:
    with open(output_file, 'w') as f:
        json.dump(combined_data, f, indent=4) # Save with indentation for readability
        print(f"Combined data saved to {output_file}. Total records: {len(combined_data)}")
except Exception as e:
    print(f"Error saving combined JSON file: {e}")
```

Combined data saved to combined_healthcare_API_Reddit_with_dates.json. Total records: 3017

Cleaning and preprocessing :

In [59]: `import json`

```
# Path to the input and output files
input_file = "cleaned_defence_API_Reddit_with_dates.json"
output_file = "2cleaned_defence_API_Reddit_with_dates.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Iterate through the entries and standardize the date field
for entry in data:
    if "date" in entry and entry["date"]:
        # Extract only the date portion (first 10 characters of the string)
        entry["date"] = entry["date"][:10]

# Save the updated JSON data to a new file
with open(output_file, "w") as file:
    json.dump(data, file, indent=4)

print(f"Dates standardized and saved to {output_file}")
```

Dates standardized and saved to 2cleaned_defence_API_Reddit_with_dates.json


```
In [66]: import json
import re

# Define a function to clean the text
def clean_text(text):
    # Decoding escaped characters (e.g., \u2019)
    text = text.encode().decode("unicode_escape")

    # Removing URLs
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)

    # Removing special characters and numbers
    text = re.sub(r'^a-zA-Z\s', '', text)

    # Converting to Lowercase
    text = text.lower()

    # Stripping Leading/trailing spaces
    text = text.strip()

    return text

# Function to preprocess the data
def preprocess_json(file_name):
    try:
        # Load the JSON file
        with open(file_name, 'r') as f:
            data = json.load(f)

        # Preprocess the data
        cleaned_data = []
        for post in data:
            cleaned_post = {}

            # Clean the title
            if 'title' in post:
                cleaned_post['title'] = clean_text(post['title'])

            # Include the score (as-is)
            if 'score' in post:
                cleaned_post['score'] = post['score']

            # Include the date (as-is)
            if 'date' in post:
                cleaned_post['date'] = post['date']

            cleaned_data.append(cleaned_post)
```

```

        return cleaned_data

    except FileNotFoundError:
        print(f"Error: {file_name} not found.")
        return []
    except json.JSONDecodeError:
        print(f"Error: Could not decode JSON in {file_name}.")
        return []

# List of JSON files to clean
json_files = [
    "combined_finance_API_Reddit_with_dates.json",
    "combined_healthcare_API_Reddit_with_dates.json",
    "2cleaned_defence_API_Reddit_with_dates.json"
]

# Clean and preprocess each file
for file_name in json_files:
    cleaned_data = preprocess_json(file_name)

    # Save the cleaned data into a new JSON file
    output_file = file_name.replace("combined", "cleaned")
    try:
        with open(output_file, 'w') as f:
            json.dump(cleaned_data, f, indent=4)
        print(f"Cleaned data saved to {output_file}. Total records: {len(cleaned_data)}")
    except Exception as e:
        print(f"Error saving cleaned JSON file: {e}")

```

Cleaned data saved to cleaned_finance_API_Reddit_with_dates.json. Total records: 3482
 Cleaned data saved to cleaned_healthcare_API_Reddit_with_dates.json. Total records: 3017
 Cleaned data saved to 2cleaned_defence_API_Reddit_with_dates.json. Total records: 4043

```
In [67]: import matplotlib.pyplot as plt
```

```
# Data for the pie chart
```

```
categories = ['Finance', 'Defence', 'Healthcare']
```

```
records = [3482, 4043, 3017]
```

```
total_records = sum(records)
```

```
# Create the pie chart
```

```
plt.figure(figsize=(8, 8))
```

```
colors = ['orangered', 'crimson', 'darkcyan']
```

```
explode = (0.02, 0.02, 0.02) # Slightly separate the slices for emphasis
```

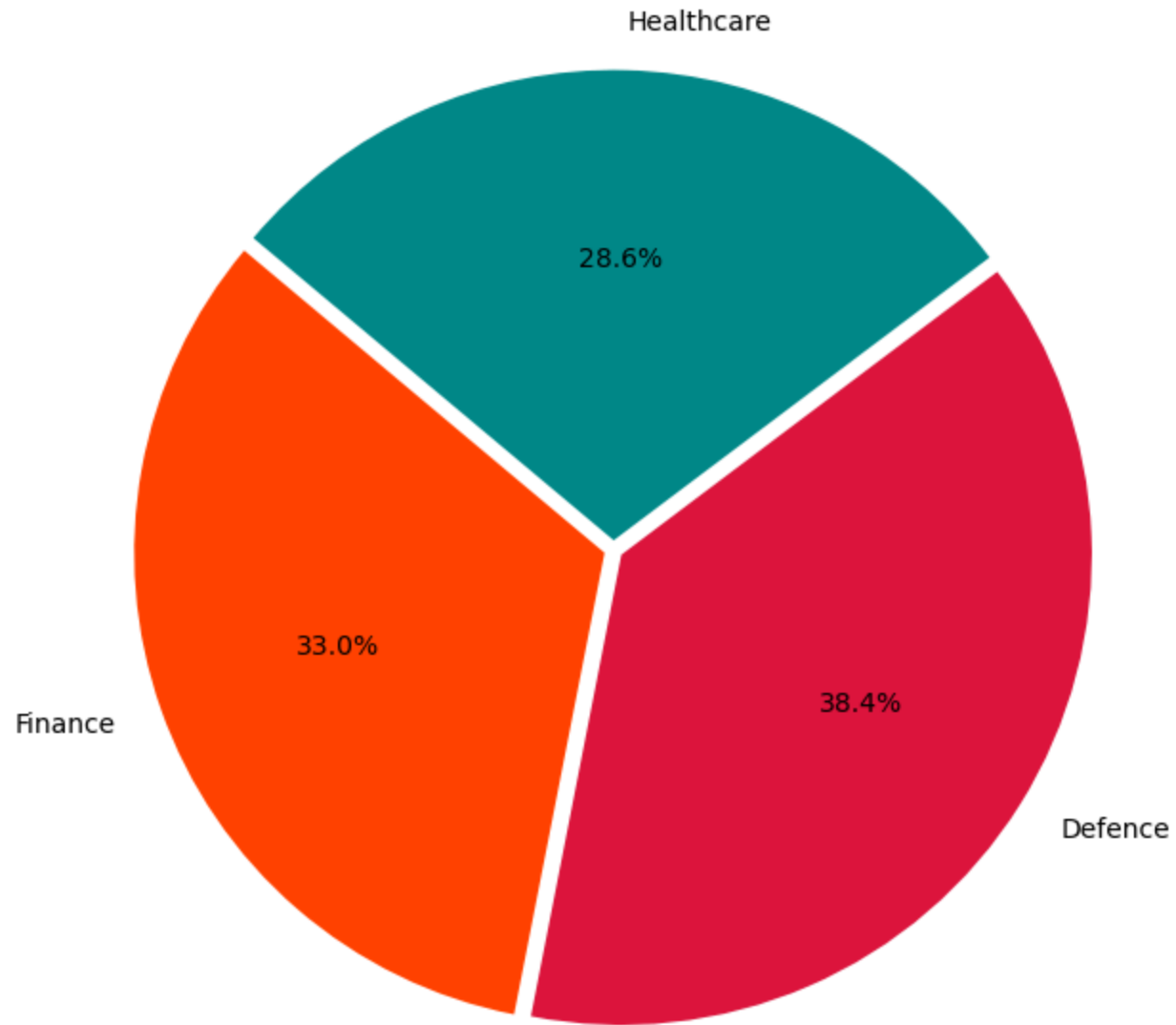
```
plt.pie(records, labels=categories, autopct='%1.1f%%', startangle=140, colors=colors, explode=explode)
```

```
# Add title with total records
```

```
plt.title(f'Distribution of Cleaned Records Fetched Using APIs\n(Total Records: {total_records})', fontsize=14)
```

```
plt.show()
```

Distribution of Cleaned Records Fetched Using APIs
(Total Records: 10542)




```
In [37]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics to search for related to Trump's win and its impact on the financial sector in the USA
topics = [
    "Trump's Election Impact on U.S. Financial Markets",
    "Trump's Victory and Wall Street Reactions",
    "Post-Trump Election Financial Policies",
    "Trump's Win and U.S. Stock Market Trends",
    "Trump's Election and Financial Sector Outlook",
    "Trump's Victory and U.S. Economic Policies",
    "Trump's Win and U.S. Financial Regulations",
    "Trump's Election and U.S. Banking Sector",
    "Trump's Victory and U.S. Investment Strategies",
    "Trump's Win and U.S. Financial News",
    "Americans' Reactions to Trump's Win",
    "Trump's Win and American Public Sentiment",
    "Social Media Reactions to Trump's Victory",
    "Trump's Win and U.S. Voter Reactions",
    "How Social Media Reacted to Trump's Election",
    "Americans Discussing Trump on Social Media",
    "Public Opinion on Trump's Victory",
    "Trump's Election and Social Media Buzz",
    "Social Media Sentiment on Trump's Win",
    "Americans' Feelings After Trump's Victory",
    "Trump's Win and American Emotions on Social Media",
    "Trump's Election: What Do Americans Really Think?",
    "Public Sentiment on Trump's Victory in 2024",
    "Social Media's Response to Trump Winning Again",
    "Trump's Win and Its Impact on American Society",
    "How Trump's Election Affected U.S. Public Opinion",
    "Trump's Victory: Divided Reactions Among Americans",
    "Social Media Discussions on Trump's Second Term",
    "Trump's Win and the Future of U.S. Politics",
    "Reactions to Trump's Election on Twitter and Facebook",
    "How Did Trump's Win Affect American Families?",
    "Trump's Presidency: Public Views and Media Reaction",
    "American Reactions to Trump's Policies After Winning",
    "Trump's Victory and Its Effect on Social Media Trends",
    "U.S. Voters' Opinions on Trump's Election Win",
    "Trump Win Reactions: What Americans Are Saying Online",
    "Trump's Win and America's Political Divide",
    "Public Opinion on Trump's Policies Post-Election",
```

```

    "Trump's Impact on U.S. Economy: Public Perception",
    "Trump Administration"
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

    while len(all_articles) < total_articles:
        params = {
            "q": topic, # Query keyword
            "apiKey": API_KEY, # API Key
            "language": "en", # Language preference
            "sortBy": "relevance", # Sort articles by relevance
            "pageSize": page_size, # Number of articles per request
            "page": page # Pagination page number
        }

        response = requests.get(BASE_URL, params=params)

        if response.status_code == 200:
            articles = response.json()["articles"]
            all_articles.extend(articles)
            print(f"Fetchd {len(articles)} articles from page {page}")
            page += 1
        else:
            print(f"Error fetching news for topic '{topic}': {response.status_code}")
            break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file

```

```
with open("finance8_API_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")
```

Fetching news for: Trump's Election Impact on U.S. Financial Markets
Fetched 44 articles from page 1
Error fetching news for topic 'Trump's Election Impact on U.S. Financial Markets': 426
Total articles fetched for 'Trump's Election Impact on U.S. Financial Markets': 44

Fetching news for: Trump's Victory and Wall Street Reactions
Fetched 5 articles from page 1
Error fetching news for topic 'Trump's Victory and Wall Street Reactions': 426
Total articles fetched for 'Trump's Victory and Wall Street Reactions': 5

Fetching news for: Post-Trump Election Financial Policies
Fetched 100 articles from page 1
Error fetching news for topic 'Post-Trump Election Financial Policies': 426
Total articles fetched for 'Post-Trump Election Financial Policies': 100

Fetching news for: Trump's Win and U.S. Stock Market Trends
Fetched 9 articles from page 1
Error fetching news for topic 'Trump's Win and U.S. Stock Market Trends': 426
Total articles fetched for 'Trump's Win and U.S. Stock Market Trends': 9

Fetching news for: Trump's Election and Financial Sector Outlook
Fetched 20 articles from page 1
Error fetching news for topic 'Trump's Election and Financial Sector Outlook': 426
Total articles fetched for 'Trump's Election and Financial Sector Outlook': 20

Fetching news for: Trump's Victory and U.S. Economic Policies
Fetched 84 articles from page 1
Error fetching news for topic 'Trump's Victory and U.S. Economic Policies': 426
Total articles fetched for 'Trump's Victory and U.S. Economic Policies': 84

Fetching news for: Trump's Win and U.S. Financial Regulations
Fetched 52 articles from page 1
Error fetching news for topic 'Trump's Win and U.S. Financial Regulations': 426
Total articles fetched for 'Trump's Win and U.S. Financial Regulations': 52

Fetching news for: Trump's Election and U.S. Banking Sector
Fetched 29 articles from page 1
Error fetching news for topic 'Trump's Election and U.S. Banking Sector': 426
Total articles fetched for 'Trump's Election and U.S. Banking Sector': 29

Fetching news for: Trump's Victory and U.S. Investment Strategies
Fetched 4 articles from page 1
Error fetching news for topic 'Trump's Victory and U.S. Investment Strategies': 426
Total articles fetched for 'Trump's Victory and U.S. Investment Strategies': 4

Fetching news for: Trump's Win and U.S. Financial News

Fetched 97 articles from page 1
 Error fetching news for topic 'Trump's Win and U.S. Financial News': 426
 Total articles fetched for 'Trump's Win and U.S. Financial News': 97

Fetching news for: Americans' Reactions to Trump's Win
 Fetched 11 articles from page 1
 Error fetching news for topic 'Americans' Reactions to Trump's Win': 426
 Total articles fetched for 'Americans' Reactions to Trump's Win': 11

Fetching news for: Trump's Win and American Public Sentiment
 Fetched 9 articles from page 1
 Error fetching news for topic 'Trump's Win and American Public Sentiment': 426
 Total articles fetched for 'Trump's Win and American Public Sentiment': 9

Fetching news for: Social Media Reactions to Trump's Victory
 Fetched 9 articles from page 1
 Error fetching news for topic 'Social Media Reactions to Trump's Victory': 426
 Total articles fetched for 'Social Media Reactions to Trump's Victory': 9

Fetching news for: Trump's Win and U.S. Voter Reactions
 Fetched 0 articles from page 1
 Error fetching news for topic 'Trump's Win and U.S. Voter Reactions': 426
 No articles found for 'Trump's Win and U.S. Voter Reactions'.

Fetching news for: How Social Media Reacted to Trump's Election
 Fetched 14 articles from page 1
 Error fetching news for topic 'How Social Media Reacted to Trump's Election': 426
 Total articles fetched for 'How Social Media Reacted to Trump's Election': 14

Fetching news for: Americans Discussing Trump on Social Media
 Fetched 57 articles from page 1
 Error fetching news for topic 'Americans Discussing Trump on Social Media': 426
 Total articles fetched for 'Americans Discussing Trump on Social Media': 57

Fetching news for: Public Opinion on Trump's Victory
 Fetched 71 articles from page 1
 Error fetching news for topic 'Public Opinion on Trump's Victory': 426
 Total articles fetched for 'Public Opinion on Trump's Victory': 71

Fetching news for: Trump's Election and Social Media Buzz
 Fetched 12 articles from page 1
 Error fetching news for topic 'Trump's Election and Social Media Buzz': 426
 Total articles fetched for 'Trump's Election and Social Media Buzz': 12

Fetching news for: Social Media Sentiment on Trump's Win
 Fetched 18 articles from page 1
 Error fetching news for topic 'Social Media Sentiment on Trump's Win': 426
 Total articles fetched for 'Social Media Sentiment on Trump's Win': 18

Fetching news for: Americans' Feelings After Trump's Victory

Fetches 8 articles from page 1

Error fetching news for topic 'Americans' Feelings After Trump's Victory': 426

Total articles fetched for 'Americans' Feelings After Trump's Victory': 8

Fetching news for: Trump's Win and American Emotions on Social Media

Fetches 3 articles from page 1

Error fetching news for topic 'Trump's Win and American Emotions on Social Media': 426

Total articles fetched for 'Trump's Win and American Emotions on Social Media': 3

Fetching news for: Trump's Election: What Do Americans Really Think?

Fetches 18 articles from page 1

Error fetching news for topic 'Trump's Election: What Do Americans Really Think?': 426

Total articles fetched for 'Trump's Election: What Do Americans Really Think?': 18

Fetching news for: Public Sentiment on Trump's Victory in 2024

Fetches 16 articles from page 1

Error fetching news for topic 'Public Sentiment on Trump's Victory in 2024': 426

Total articles fetched for 'Public Sentiment on Trump's Victory in 2024': 16

Fetching news for: Social Media's Response to Trump Winning Again

Fetches 0 articles from page 1

Error fetching news for topic 'Social Media's Response to Trump Winning Again': 426

No articles found for 'Social Media's Response to Trump Winning Again'.

Fetching news for: Trump's Win and Its Impact on American Society

Fetches 11 articles from page 1

Error fetching news for topic 'Trump's Win and Its Impact on American Society': 426

Total articles fetched for 'Trump's Win and Its Impact on American Society': 11

Fetching news for: How Trump's Election Affected U.S. Public Opinion

Fetches 1 articles from page 1

Error fetching news for topic 'How Trump's Election Affected U.S. Public Opinion': 426

Total articles fetched for 'How Trump's Election Affected U.S. Public Opinion': 1

Fetching news for: Trump's Victory: Divided Reactions Among Americans

Fetches 0 articles from page 1

Error fetching news for topic 'Trump's Victory: Divided Reactions Among Americans': 426

No articles found for 'Trump's Victory: Divided Reactions Among Americans'.

Fetching news for: Social Media Discussions on Trump's Second Term

Fetches 44 articles from page 1

Error fetching news for topic 'Social Media Discussions on Trump's Second Term': 426

Total articles fetched for 'Social Media Discussions on Trump's Second Term': 44

Fetching news for: Trump's Win and the Future of U.S. Politics

Fetches 37 articles from page 1

Error fetching news for topic 'Trump's Win and the Future of U.S. Politics': 426
Total articles fetched for 'Trump's Win and the Future of U.S. Politics': 37

Fetching news for: Reactions to Trump's Election on Twitter and Facebook
Fetched 3 articles from page 1
Error fetching news for topic 'Reactions to Trump's Election on Twitter and Facebook': 426
Total articles fetched for 'Reactions to Trump's Election on Twitter and Facebook': 3

Fetching news for: How Did Trump's Win Affect American Families?
Fetched 0 articles from page 1
Error fetching news for topic 'How Did Trump's Win Affect American Families?': 426
No articles found for 'How Did Trump's Win Affect American Families?'.

Fetching news for: Trump's Presidency: Public Views and Media Reaction
Fetched 4 articles from page 1
Error fetching news for topic 'Trump's Presidency: Public Views and Media Reaction': 426
Total articles fetched for 'Trump's Presidency: Public Views and Media Reaction': 4

Fetching news for: American Reactions to Trump's Policies After Winning
Fetched 0 articles from page 1
Error fetching news for topic 'American Reactions to Trump's Policies After Winning': 426
No articles found for 'American Reactions to Trump's Policies After Winning'.

Fetching news for: Trump's Victory and Its Effect on Social Media Trends

```

    Fetched 0 articles from page 1
    Error fetching news for topic 'Trump's Victory and Its Effect on Social Media Trends': 426
    No articles found for 'Trump's Victory and Its Effect on Social Media Trends'.

```

```

    Fetching news for: U.S. Voters' Opinions on Trump's Election Win
    Fetched 9 articles from page 1
    Error fetching news for topic 'U.S. Voters' Opinions on Trump's Election Win': 426
    Total articles fetched for 'U.S. Voters' Opinions on Trump's Election Win': 9

```

```

    Fetching news for: Trump Win Reactions: What Americans Are Saying Online
    Fetched 13 articles from page 1
    Error fetching news for topic 'Trump Win Reactions: What Americans Are Saying Online': 426
    Total articles fetched for 'Trump Win Reactions: What Americans Are Saying Online': 13

```

```

    Fetching news for: Trump's Win and America's Political Divide
    Fetched 1 articles from page 1
    Error fetching news for topic 'Trump's Win and America's Political Divide': 426
    Total articles fetched for 'Trump's Win and America's Political Divide': 1

```

```

    Fetching news for: Public Opinion on Trump's Policies Post-Election
    Fetched 30 articles from page 1
    Error fetching news for topic 'Public Opinion on Trump's Policies Post-Election': 426
    Total articles fetched for 'Public Opinion on Trump's Policies Post-Election': 30

```

```

    Fetching news for: Trump's Impact on U.S. Economy: Public Perception
    Fetched 2 articles from page 1
    Error fetching news for topic 'Trump's Impact on U.S. Economy: Public Perception': 426
    Total articles fetched for 'Trump's Impact on U.S. Economy: Public Perception': 2

```

```

    Fetching news for: Trump Administration
    Fetched 100 articles from page 1
    Error fetching news for topic 'Trump Administration': 426
    Total articles fetched for 'Trump Administration': 100

```

```

    Total number of articles fetched: 945

```

In []:


```

In [51]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# More simple and effective topics related to Trump's win, social media, and American sentiment
topics = [
    "Trump's Victory and Public Opinion",
    "How Americans Feel About Trump's Election",
    "Trump's Win on Social Media",
    "Trump Election Reactions on Twitter",
    "Trump's Second Term and U.S. Reactions",
    "Americans Celebrate Trump's Win",
    "Trump's Impact on U.S. Society",
    "Trump's Election: What People Are Saying",
    "Social Media Buzz Around Trump's Win",
    "Trump's Win and U.S. Citizens' Views",
    "Trump's Victory and American Politics",
    "How Americans Are Reacting to Trump's Victory",
    "Trump's Impact on American Families",
    "Trump's Victory and Its Effect on Jobs",
    "Trump's Election and Financial Markets Reaction",
    "Trump's Win: Social Media and the People",
    "Americans Discuss Trump's Second Term",
    "Trump and U.S. Economy After His Win",
    "What Americans Think About Trump's Win",
    "Trump's Impact on American Jobs and Economy"
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

    while len(all_articles) < total_articles:
        params = {
            "q": topic, # Query keyword
            "apiKey": API_KEY, # API Key
            "language": "en", # Language preference
            "sortBy": "relevance", # Sort articles by relevance
            "pageSize": page_size, # Number of articles per request
            "page": page # Pagination page number

```

```

    }

    response = requests.get(BASE_URL, params=params)

    if response.status_code == 200:
        articles = response.json()["articles"]
        all_articles.extend(articles)
        print(f"Fetched {len(articles)} articles from page {page}")
        page += 1
    else:
        print(f"Error fetching news for topic '{topic}': {response.status_code}")
        break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("finance9_API_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")

```

Fetching news for: Trump's Victory and Public Opinion
Fetched 71 articles from page 1
Error fetching news for topic 'Trump's Victory and Public Opinion': 426
Total articles fetched for 'Trump's Victory and Public Opinion': 71

Fetching news for: How Americans Feel About Trump's Election
Fetched 86 articles from page 1
Error fetching news for topic 'How Americans Feel About Trump's Election': 426
Total articles fetched for 'How Americans Feel About Trump's Election': 86

Fetching news for: Trump's Win on Social Media
Fetched 100 articles from page 1
Error fetching news for topic 'Trump's Win on Social Media': 426
Total articles fetched for 'Trump's Win on Social Media': 100

Fetching news for: Trump Election Reactions on Twitter
Fetched 31 articles from page 1
Error fetching news for topic 'Trump Election Reactions on Twitter': 426
Total articles fetched for 'Trump Election Reactions on Twitter': 31

Fetching news for: Trump's Second Term and U.S. Reactions
Fetched 14 articles from page 1
Error fetching news for topic 'Trump's Second Term and U.S. Reactions': 426
Total articles fetched for 'Trump's Second Term and U.S. Reactions': 14

Fetching news for: Americans Celebrate Trump's Win
Fetched 11 articles from page 1
Error fetching news for topic 'Americans Celebrate Trump's Win': 426
Total articles fetched for 'Americans Celebrate Trump's Win': 11

Fetching news for: Trump's Impact on U.S. Society
Fetched 43 articles from page 1
Error fetching news for topic 'Trump's Impact on U.S. Society': 426
Total articles fetched for 'Trump's Impact on U.S. Society': 43

Fetching news for: Trump's Election: What People Are Saying
Fetched 100 articles from page 1
Error fetching news for topic 'Trump's Election: What People Are Saying': 426
Total articles fetched for 'Trump's Election: What People Are Saying': 100

Fetching news for: Social Media Buzz Around Trump's Win
Fetched 0 articles from page 1
Error fetching news for topic 'Social Media Buzz Around Trump's Win': 426
No articles found for 'Social Media Buzz Around Trump's Win'.

Fetching news for: Trump's Win and U.S. Citizens' Views

Fetched 6 articles from page 1
 Error fetching news for topic 'Trump's Win and U.S. Citizens' Views': 426
 Total articles fetched for 'Trump's Win and U.S. Citizens' Views': 6

Fetching news for: Trump's Victory and American Politics
 Error fetching news for topic 'Trump's Victory and American Politics': 429
 No articles found for 'Trump's Victory and American Politics'.

Fetching news for: How Americans Are Reacting to Trump's Victory
 Error fetching news for topic 'How Americans Are Reacting to Trump's Victory': 429
 No articles found for 'How Americans Are Reacting to Trump's Victory'.

Fetching news for: Trump's Impact on American Families
 Error fetching news for topic 'Trump's Impact on American Families': 429
 No articles found for 'Trump's Impact on American Families'.

Fetching news for: Trump's Victory and Its Effect on Jobs
 Error fetching news for topic 'Trump's Victory and Its Effect on Jobs': 429
 No articles found for 'Trump's Victory and Its Effect on Jobs'.

Fetching news for: Trump's Election and Financial Markets Reaction
 Error fetching news for topic 'Trump's Election and Financial Markets Reaction': 429
 No articles found for 'Trump's Election and Financial Markets Reaction'.

Fetching news for: Trump's Win: Social Media and the People
 Error fetching news for topic 'Trump's Win: Social Media and the People': 429
 No articles found for 'Trump's Win: Social Media and the People'.

Fetching news for: Americans Discuss Trump's Second Term
 Error fetching news for topic 'Americans Discuss Trump's Second Term': 429
 No articles found for 'Americans Discuss Trump's Second Term'.

Fetching news for: Trump and U.S. Economy After His Win
 Error fetching news for topic 'Trump and U.S. Economy After His Win': 429
 No articles found for 'Trump and U.S. Economy After His Win'.

Fetching news for: What Americans Think About Trump's Win
 Error fetching news for topic 'What Americans Think About Trump's Win': 429
 No articles found for 'What Americans Think About Trump's Win'.

Fetching news for: Trump's Impact on American Jobs and Economy
 Error fetching news for topic 'Trump's Impact on American Jobs and Economy': 429
 No articles found for 'Trump's Impact on American Jobs and Economy'.

Total number of articles fetched: 462


```

In [57]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# More simple and effective topics related to Trump's win, social media, and American sentiment
# More topics related to the U.S. finance sector
topics = [
    "U.S. Stock Market After Trump's Win",
    "Impact of Trump's Election on U.S. Banks",
    "Trump's Policies and U.S. Financial Growth",
    "How Trump's Win Affects Wall Street",
    "Trump's Victory and U.S. Financial Markets",
    "Trump's Effect on U.S. Banking Sector",
    "Finance Trends in the U.S. Post-Trump",
    "U.S. Economic Policies After Trump's Win",
    "Trump's Impact on U.S. Investment Market",
    "Financial Strategies in the U.S. After Trump's Win",
    "Stock Market Predictions After Trump's Election",
    "Trump's Tax Policies and the U.S. Economy",
    "Trump's Economic Plan and U.S. Financial Growth",
    "U.S. Banks' Responses to Trump's Policies",
    "U.S. Economy: What Trump's Win Means for Finance",
    "Trump's Financial Regulations and Wall Street",
    "Trump's Influence on U.S. Investment Trends",
    "How Trump's Policies Affect U.S. Stock Investments",
    "U.S. Real Estate Market Post-Trump Election",
    "Impact of Trump's Financial Policies on Investors"
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

    while len(all_articles) < total_articles:
        params = {
            "q": topic, # Query keyword
            "apiKey": API_KEY, # API Key
            "language": "en", # Language preference
            "sortBy": "relevance", # Sort articles by relevance
            "pageSize": page_size, # Number of articles per request

```

```

        "page": page                # Pagination page number
    }

    response = requests.get(BASE_URL, params=params)

    if response.status_code == 200:
        articles = response.json()["articles"]
        all_articles.extend(articles)
        print(f"Fetched {len(articles)} articles from page {page}")
        page += 1
    else:
        print(f"Error fetching news for topic '{topic}': {response.status_code}")
        break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("finance10_API_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")

```


Fetching news for: U.S. Stock Market After Trump's Win
Error fetching news for topic 'U.S. Stock Market After Trump's Win': 429
No articles found for 'U.S. Stock Market After Trump's Win'.

Fetching news for: Impact of Trump's Election on U.S. Banks
Error fetching news for topic 'Impact of Trump's Election on U.S. Banks': 429
No articles found for 'Impact of Trump's Election on U.S. Banks'.

Fetching news for: Trump's Policies and U.S. Financial Growth
Error fetching news for topic 'Trump's Policies and U.S. Financial Growth': 429
No articles found for 'Trump's Policies and U.S. Financial Growth'.

Fetching news for: How Trump's Win Affects Wall Street
Error fetching news for topic 'How Trump's Win Affects Wall Street': 429
No articles found for 'How Trump's Win Affects Wall Street'.

Fetching news for: Trump's Victory and U.S. Financial Markets
Error fetching news for topic 'Trump's Victory and U.S. Financial Markets': 429
No articles found for 'Trump's Victory and U.S. Financial Markets'.

Fetching news for: Trump's Effect on U.S. Banking Sector
Error fetching news for topic 'Trump's Effect on U.S. Banking Sector': 429
No articles found for 'Trump's Effect on U.S. Banking Sector'.

Fetching news for: Finance Trends in the U.S. Post-Trump
Error fetching news for topic 'Finance Trends in the U.S. Post-Trump': 429
No articles found for 'Finance Trends in the U.S. Post-Trump'.

Fetching news for: U.S. Economic Policies After Trump's Win
Error fetching news for topic 'U.S. Economic Policies After Trump's Win': 429
No articles found for 'U.S. Economic Policies After Trump's Win'.

Fetching news for: Trump's Impact on U.S. Investment Market
Error fetching news for topic 'Trump's Impact on U.S. Investment Market': 429
No articles found for 'Trump's Impact on U.S. Investment Market'.

Fetching news for: Financial Strategies in the U.S. After Trump's Win
Error fetching news for topic 'Financial Strategies in the U.S. After Trump's Win': 429
No articles found for 'Financial Strategies in the U.S. After Trump's Win'.

Fetching news for: Stock Market Predictions After Trump's Election
Error fetching news for topic 'Stock Market Predictions After Trump's Election': 429
No articles found for 'Stock Market Predictions After Trump's Election'.

Fetching news for: Trump's Tax Policies and the U.S. Economy
Error fetching news for topic 'Trump's Tax Policies and the U.S. Economy': 429

No articles found for 'Trump's Tax Policies and the U.S. Economy'.

Fetching news for: Trump's Economic Plan and U.S. Financial Growth

Error fetching news for topic 'Trump's Economic Plan and U.S. Financial Growth': 429

No articles found for 'Trump's Economic Plan and U.S. Financial Growth'.

Fetching news for: U.S. Banks' Responses to Trump's Policies

Error fetching news for topic 'U.S. Banks' Responses to Trump's Policies': 429

No articles found for 'U.S. Banks' Responses to Trump's Policies'.

Fetching news for: U.S. Economy: What Trump's Win Means for Finance

Error fetching news for topic 'U.S. Economy: What Trump's Win Means for Finance': 429

No articles found for 'U.S. Economy: What Trump's Win Means for Finance'.

Fetching news for: Trump's Financial Regulations and Wall Street

Error fetching news for topic 'Trump's Financial Regulations and Wall Street': 429

No articles found for 'Trump's Financial Regulations and Wall Street'.

Fetching news for: Trump's Influence on U.S. Investment Trends

Error fetching news for topic 'Trump's Influence on U.S. Investment Trends': 429

No articles found for 'Trump's Influence on U.S. Investment Trends'.

Fetching news for: How Trump's Policies Affect U.S. Stock Investments

Error fetching news for topic 'How Trump's Policies Affect U.S. Stock Investments': 429

No articles found for 'How Trump's Policies Affect U.S. Stock Investments'.

Fetching news for: U.S. Real Estate Market Post-Trump Election

Error fetching news for topic 'U.S. Real Estate Market Post-Trump Election': 429

No articles found for 'U.S. Real Estate Market Post-Trump Election'.

Fetching news for: Impact of Trump's Financial Policies on Investors

Error fetching news for topic 'Impact of Trump's Financial Policies on Investors': 429

No articles found for 'Impact of Trump's Financial Policies on Investors'.

Total number of articles fetched: 0

In [13]: !pip install requests

Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\site-packages (2.31.0)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\admin\anaconda3\lib\site-packages (from requests) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests) (1.26.16)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests) (2024.8.30)


```

In [17]: import requests
import json
from datetime import datetime

# Your GNews API key
api_key = "1bd88a87e8c400204840189f8484382e"

# Base URL for GNews API
base_url = "https://gnews.io/api/v4/search"

topics = [
    "Trump economic policies",
    "Trump tax reforms",
    "Trump trade wars and finance",
    "Trump tariffs impact on economy",
    "Trump financial regulations",
    "Trump's Wall Street policies",
    "Trump's stock market influence",
    "Trump 2024 financial policies",
    "Trump's tax cuts and the economy",
    "Impact of Trump's win on US markets",
    "Trump policies and financial markets",
    "US economy after Trump election",
    "Trump's impact on banking sector",
    "Trump policies and investor sentiment",
    "Trump's influence on US dollar",
    "Trump's fiscal policies and debt",
    "Trump's role in corporate tax reform",
    "Trump's effect on trade relations and finance",
    "Trump's approach to financial regulation rollback",
    "Trump's economic legacy on US finance"
]

# Function to fetch news articles for a specific topic with pagination
def fetch_news(topic, page):
    params = {
        'q': topic,
        'token': api_key,
        'lang': 'en', # Search in English
        'sort': 'published_at', # Sort by publication date
        'max': 100, # Set the max number of articles per request
        'page': page # Page number for pagination
    }

    response = requests.get(base_url, params=params)

```

```

# If the request was successful
if response.status_code == 200:
    return response.json()
else:
    print(f"Error fetching news for {topic}. Status Code: {response.status_code}")
    return None

# Function to get all news articles for a specific topic
def get_all_articles_for_topic(topic):
    all_articles = []
    page = 1

    while True:
        news_data = fetch_news(topic, page)

        if news_data and news_data['articles']:
            all_articles.extend(news_data['articles'])
            page += 1 # Move to the next page
        else:
            break # No more articles to fetch

    return all_articles

# Fetch news for each topic and save results in a list
all_articles = []

for topic in topics:
    print(f"Fetching all news articles for topic: {topic}")
    articles = get_all_articles_for_topic(topic)
    # Add timestamp and publication date to each article
    for article in articles:
        article['fetched_at'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S') # Current fetch time
        article['published_at'] = article.get('publishedAt') # Extract published date from the API response
    all_articles.extend(articles)

# Save the collected news data into a JSON file
with open('finance11_API_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

print("All news articles with dates saved successfully!")

```

Fetching all news articles for topic: Trump economic policies
Error fetching news for Trump economic policies. Status Code: 429
Fetching all news articles for topic: Trump tax reforms
Error fetching news for Trump tax reforms. Status Code: 429
Fetching all news articles for topic: Trump trade wars and finance
Error fetching news for Trump trade wars and finance. Status Code: 429
Fetching all news articles for topic: Trump tariffs impact on economy
Error fetching news for Trump tariffs impact on economy. Status Code: 429
Fetching all news articles for topic: Trump financial regulations
Error fetching news for Trump financial regulations. Status Code: 429
Fetching all news articles for topic: Trump's Wall Street policies
Error fetching news for Trump's Wall Street policies. Status Code: 400
Fetching all news articles for topic: Trump's stock market influence
Error fetching news for Trump's stock market influence. Status Code: 400
Fetching all news articles for topic: Trump 2024 financial policies
Fetching all news articles for topic: Trump's tax cuts and the economy
Error fetching news for Trump's tax cuts and the economy. Status Code: 429
Fetching all news articles for topic: Impact of Trump's win on US markets
Error fetching news for Impact of Trump's win on US markets. Status Code: 400
Fetching all news articles for topic: Trump policies and financial markets
Error fetching news for Trump policies and financial markets. Status Code: 429
Fetching all news articles for topic: US economy after Trump election
Error fetching news for US economy after Trump election. Status Code: 429
Fetching all news articles for topic: Trump's impact on banking sector
Error fetching news for Trump's impact on banking sector. Status Code: 400
Fetching all news articles for topic: Trump policies and investor sentiment
Error fetching news for Trump policies and investor sentiment. Status Code: 429
Fetching all news articles for topic: Trump's influence on US dollar
Error fetching news for Trump's influence on US dollar. Status Code: 400
Fetching all news articles for topic: Trump's fiscal policies and debt
Error fetching news for Trump's fiscal policies and debt. Status Code: 400
Fetching all news articles for topic: Trump's role in corporate tax reform
Error fetching news for Trump's role in corporate tax reform. Status Code: 400
Fetching all news articles for topic: Trump's effect on trade relations and finance
Error fetching news for Trump's effect on trade relations and finance. Status Code: 429
Fetching all news articles for topic: Trump's approach to financial regulation rollback
Error fetching news for Trump's approach to financial regulation rollback. Status Code: 400
Fetching all news articles for topic: Trump's economic legacy on US finance
Error fetching news for Trump's economic legacy on US finance. Status Code: 400
All news articles with dates saved successfully!

In [22]: `import json`

```
# Path to the saved JSON file
file_path = 'finance13_API_with_dates.json'

# Load the JSON data from the file
with open(file_path, 'r') as file:
    data = json.load(file)

# Assuming that the articles are stored in a list in the JSON file
total_articles = len(data)

# Print the total number of articles
print(f"Total number of articles fetched: {total_articles}")
```

Total number of articles fetched: 60


```

In [19]: import requests
import json
from datetime import datetime

# Your GNews API key
api_key = "1bd88a87e8c400204840189f8484382e"

# Base URL for GNews API
base_url = "https://gnews.io/api/v4/search"

topics = [
    "US Stock Market After Trump Win",
    "Trump Election Impact on US Banks",
    "Trump Policies and US Economy",
    "Trump Win and US Financial Markets",
    "US Economy After Trump Win",
    "Trump Influence on Wall Street",
    "Trump's Economic Impact on Banks",
    "US Financial Growth After Trump",
    "Trump Tax Cuts and the Economy",
    "Trump Effect on US Investment Market",
    "Post-Trump Stock Market Trends",
    "Trump's Economic Policies and Finance",
    "Trump's Tax Reform and US Economy",
    "US Banks under Trump's Policies",
    "Trump's Economic Plan and US Growth",
    "Trump's Effect on US Financial Sector",
    "Trump's Policies and US Investments",
    "US Real Estate Market After Trump",
    "Trump's Financial Policies and Investors",
    "US Economic Outlook After Trump Win",
    "Trump and the US Investment Market"
]

# Function to fetch news articles for a specific topic with pagination
def fetch_news(topic, page):
    params = {
        'q': topic,
        'token': api_key,
        'lang': 'en', # Search in English
        'sort': 'published_at', # Sort by publication date
        'max': 100, # Set the max number of articles per request
        'page': page # Page number for pagination
    }

```

```

response = requests.get(base_url, params=params)

# If the request was successful
if response.status_code == 200:
    return response.json()
else:
    print(f"Error fetching news for {topic}. Status Code: {response.status_code}")
    return None

# Function to get all news articles for a specific topic
def get_all_articles_for_topic(topic):
    all_articles = []
    page = 1

    while True:
        news_data = fetch_news(topic, page)

        if news_data and news_data['articles']:
            all_articles.extend(news_data['articles'])
            page += 1 # Move to the next page
        else:
            break # No more articles to fetch

    return all_articles

# Fetch news for each topic and save results in a list
all_articles = []

for topic in topics:
    print(f"Fetching all news articles for topic: {topic}")
    articles = get_all_articles_for_topic(topic)
    # Add timestamp and publication date to each article
    for article in articles:
        article['fetched_at'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S') # Current fetch time
        article['published_at'] = article.get('publishedAt') # Extract published date from the API response
    all_articles.extend(articles)

# Save the collected news data into a JSON file
with open('finance12_API_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

print("All news articles with dates saved successfully!")

```

Fetching all news articles for topic: US Stock Market After Trump Win
Fetching all news articles for topic: Trump Election Impact on US Banks
Fetching all news articles for topic: Trump Policies and US Economy
Error fetching news for Trump Policies and US Economy. Status Code: 429
Fetching all news articles for topic: Trump Win and US Financial Markets
Error fetching news for Trump Win and US Financial Markets. Status Code: 429
Fetching all news articles for topic: US Economy After Trump Win
Error fetching news for US Economy After Trump Win. Status Code: 429
Fetching all news articles for topic: Trump Influence on Wall Street
Error fetching news for Trump Influence on Wall Street. Status Code: 429
Fetching all news articles for topic: Trump's Economic Impact on Banks
Error fetching news for Trump's Economic Impact on Banks. Status Code: 400
Fetching all news articles for topic: US Financial Growth After Trump
Fetching all news articles for topic: Trump Tax Cuts and the Economy
Error fetching news for Trump Tax Cuts and the Economy. Status Code: 429
Fetching all news articles for topic: Trump Effect on US Investment Market
Fetching all news articles for topic: Post-Trump Stock Market Trends
Error fetching news for Post-Trump Stock Market Trends. Status Code: 400
Fetching all news articles for topic: Trump's Economic Policies and Finance
Error fetching news for Trump's Economic Policies and Finance. Status Code: 400
Fetching all news articles for topic: Trump's Tax Reform and US Economy
Error fetching news for Trump's Tax Reform and US Economy. Status Code: 429
Fetching all news articles for topic: US Banks under Trump's Policies
Error fetching news for US Banks under Trump's Policies. Status Code: 400
Fetching all news articles for topic: Trump's Economic Plan and US Growth
Error fetching news for Trump's Economic Plan and US Growth. Status Code: 400
Fetching all news articles for topic: Trump's Effect on US Financial Sector
Error fetching news for Trump's Effect on US Financial Sector. Status Code: 400
Fetching all news articles for topic: Trump's Policies and US Investments
Error fetching news for Trump's Policies and US Investments. Status Code: 429
Fetching all news articles for topic: US Real Estate Market After Trump
Fetching all news articles for topic: Trump's Financial Policies and Investors
Error fetching news for Trump's Financial Policies and Investors. Status Code: 400
Fetching all news articles for topic: US Economic Outlook After Trump Win
Fetching all news articles for topic: Trump and the US Investment Market
Error fetching news for Trump and the US Investment Market. Status Code: 429
All news articles with dates saved successfully!


```

In [28]: import requests
import json
from datetime import datetime

# Your GNews API key
api_key = "1bd88a87e8c400204840189f8484382e"

# Base URL for GNews API
base_url = "https://gnews.io/api/v4/search"

topics = [

    "Trump and USA", "Economy of usa","dollars"

]


# Function to fetch news articles for a specific topic with pagination
def fetch_news(topic, page):
    params = {
        'q': topic,
        'token': api_key,
        'lang': 'en', # Search in English
        'sort': 'published_at', # Sort by publication date
        'max': 100, # Set the max number of articles per request
        'page': page # Page number for pagination
    }

    response = requests.get(base_url, params=params)

    # If the request was successful
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching news for {topic}. Status Code: {response.status_code}")
        return None


# Function to get all news articles for a specific topic
def get_all_articles_for_topic(topic):
    all_articles = []
    page = 1

    while True:
        news_data = fetch_news(topic, page)

```

```

        if news_data and news_data['articles']:
            all_articles.extend(news_data['articles'])
            page += 1 # Move to the next page
        else:
            break # No more articles to fetch

    return all_articles

# Fetch news for each topic and save results in a list
all_articles = []

for topic in topics:
    print(f"Fetching all news articles for topic: {topic}")
    articles = get_all_articles_for_topic(topic)
    # Add timestamp and publication date to each article
    for article in articles:
        article['fetched_at'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S') # Current fetch time
        article['published_at'] = article.get('publishedAt') # Extract published date from the API response
    all_articles.extend(articles)

# Save the collected news data into a JSON file
with open('finance14_API_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

print("All news articles with dates saved successfully!")

```

```

Fetching all news articles for topic: Trump and USA
Error fetching news for Trump and USA. Status Code: 429
Fetching all news articles for topic: Economy of usa
Error fetching news for Economy of usa. Status Code: 429
Fetching all news articles for topic: dollars
Error fetching news for dollars. Status Code: 429
All news articles with dates saved successfully!

```



```

In [35]: import requests
import json
from datetime import datetime

# Your GNews API key
api_key = "1bd88a87e8c400204840189f8484382e"

# Base URL for GNews API
base_url = "https://gnews.io/api/v4/search"

topics = [

    "Trump 2024", "America First", "Republican",

]

# Function to fetch news articles for a specific topic with pagination
def fetch_news(topic, page):
    params = {
        'q': topic,
        'token': api_key,
        'lang': 'en', # Search in English
        'sort': 'published_at', # Sort by publication date
        'max': 100, # Set the max number of articles per request
        'page': page # Page number for pagination
    }

    response = requests.get(base_url, params=params)

    # If the request was successful
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching news for {topic}. Status Code: {response.status_code}")
        return None

# Function to get all news articles for a specific topic
def get_all_articles_for_topic(topic):
    all_articles = []
    page = 1

    while True:
        news_data = fetch_news(topic, page)

```



```

        if news_data and news_data['articles']:
            all_articles.extend(news_data['articles'])
            page += 1 # Move to the next page
        else:
            break # No more articles to fetch

    return all_articles

# Fetch news for each topic and save results in a list
all_articles = []

for topic in topics:
    print(f"Fetching all news articles for topic: {topic}")
    articles = get_all_articles_for_topic(topic)
    # Add timestamp and publication date to each article
    for article in articles:
        article['fetched_at'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S') # Current fetch time
        article['published_at'] = article.get('publishedAt') # Extract published date from the API response
    all_articles.extend(articles)

# Save the collected news data into a JSON file
with open('finance15_API_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

print("All news articles with dates saved successfully!")

```

```

Fetching all news articles for topic: Trump 2024
Error fetching news for Trump 2024. Status Code: 429
Fetching all news articles for topic: America First
Error fetching news for America First. Status Code: 429
Fetching all news articles for topic: Republican
Error fetching news for Republican. Status Code: 429
All news articles with dates saved successfully!

```

In [54]: **import** json

```
# List of files to process
file_paths = ['finance11_API_with_dates.json', 'finance12_API_with_dates.json', 'finance14_API_with_dates.json',
              'finance15_API_with_dates.json',
              'finance8_list_API.json', 'finance9_list_API.json']

# Function to rename the column from 'published_at' to 'date'
def rename_published_at_column(file_path):
    # Open the JSON file and load the data
    with open(file_path, 'r') as file:
        data = json.load(file)

    # Iterate through each article and rename the column
    for article in data:
        if 'publishedAt' in article:
            article['date'] = article.pop('publishedAt')

    # Save the modified data back to the same file
    with open(file_path, 'w') as file:
        json.dump(data, file, indent=4)

# Process each file
for file_path in file_paths:
    rename_published_at_column(file_path)
    print(f"Updated '{file_path}' successfully!")
```

```
Updated 'finance11_API_with_dates.json' successfully!
Updated 'finance12_API_with_dates.json' successfully!
Updated 'finance14_API_with_dates.json' successfully!
Updated 'finance15_API_with_dates.json' successfully!
Updated 'finance8_list_API.json' successfully!
Updated 'finance9_list_API.json' successfully!
```

```
In [49]: import json

# Path to the input file
input_file = "finance8_API_with_dates.json"
output_file = "finance8_list_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Initialize a List to store processed articles
processed_articles = []

# Iterate through the topics and their articles
for topic, articles in data[0].items():
    for article in articles:
        processed_article = {
            "topic": topic,
            "source_name": article["source"].get("name"),
            "author": article.get("author"),
            "title": article.get("title"),
            "description": article.get("description"),
            "url": article.get("url"),
            "urlToImage": article.get("urlToImage"),
            "publishedAt": article.get("publishedAt"),
            "content": article.get("content"),
        }
        # Append the processed article to the List
        processed_articles.append(processed_article)

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(processed_articles, file, indent=4)

print(f"Cleaned and structured data saved to {output_file}")
```

Cleaned and structured data saved to finance8_list_API.json

In [53]: **import** json

```
# Path to the input file
input_file = "finance9_API_with_dates.json"
output_file = "finance9_list_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Print the structure of the loaded data to check the format
print(json.dumps(data, indent=4))

# Initialize a list to store processed articles
processed_articles = []

# Ensure that the data structure is a dictionary, and iterate accordingly
if isinstance(data, dict):
    # Iterate through the topics and their articles if data is a dictionary
    for topic, articles in data.items():
        for article in articles:
            processed_article = {
                "topic": topic,
                "source_name": article["source"].get("name"),
                "author": article.get("author"),
                "title": article.get("title"),
                "description": article.get("description"),
                "url": article.get("url"),
                "urlToImage": article.get("urlToImage"),
                "publishedAt": article.get("publishedAt"),
                "content": article.get("content"),
            }
            # Append the processed article to the list
            processed_articles.append(processed_article)
else:
    print("Data is not in the expected format.")

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(processed_articles, file, indent=4)

print(f"Cleaned and structured data saved to {output_file}")
```

```
{
  "Trump's Victory and Public Opinion": [
    {
      "source": {
        "id": "business-insider",
        "name": "Business Insider"
      },
      "author": "Rebecca Rommen,Nathan Rennolds",
      "title": "4 ways the war in Ukraine could play out after Trump's return to power",
      "description": "Kyiv and Moscow are racing to place their respective sides in the best possible position ahead of any changes Trump's administration may bring.",
      "url": "https://www.businessinsider.com/4-ways-war-ukraine-play-out-after-trumps-return-power-2024-12",
      "urlToImage": "https://i.insider.com/674983e1fa0140cdd56623b5?width=800&format=jpeg",
      "publishedAt": "2024-12-18T10:48:12Z",
      "content": "Trump has called the Russia-Ukraine war \"a loser\" and said he'll quickly negotiate an end to it.Chris Unger & Tetiana Dzhafarova | Getty Images\r\n<ul><li>Trump's return to power comes as Ukraine strug\u20266 [+8241 chars]"
    },
    {
      " " " "
```



```
In [66]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of topics you're interested in
topics = [
    "Trump's Election Impact on U.S. Financial Markets",
    "Trump's Victory and Wall Street Reactions",
    "Post-Trump Election Financial Policies",
    "Trump's Win and U.S. Stock Market Trends",
    "Trump's Election and Financial Sector Outlook",
    "Trump's Victory and U.S. Economic Policies",
    "Trump's Win and U.S. Financial Regulations",
    "Trump's Election and U.S. Banking Sector",
    "Trump's Victory and U.S. Investment Strategies",
    "Trump's Win and U.S. Financial News",
    "Americans' Reactions to Trump's Win",
    "Trump's Win and American Public Sentiment",
    "Social Media Reactions to Trump's Victory",
    "Trump's Win and U.S. Voter Reactions",
    "How Social Media Reacted to Trump's Election",
    "Americans Discussing Trump on Social Media",
    "Public Opinion on Trump's Victory",
    "Trump's Election and Social Media Buzz",
    "Social Media Sentiment on Trump's Win",
    "Americans' Feelings After Trump's Victory",
    "Trump's Win and American Emotions on Social Media",
    "Trump's Election: What Do Americans Really Think?",
    "Public Sentiment on Trump's Victory in 2024",
    "Social Media's Response to Trump Winning Again",
    "Trump's Win and Its Impact on American Society",
    "How Trump's Election Affected U.S. Public Opinion",
    "Trump's Victory: Divided Reactions Among Americans",
    "Social Media Discussions on Trump's Second Term",
    "Trump's Win and the Future of U.S. Politics",
    "Reactions to Trump's Election on Twitter and Facebook",
    "How Did Trump's Win Affect American Families?",
    "Trump's Presidency: Public Views and Media Reaction",
    "American Reactions to Trump's Policies After Winning",
    "Trump's Victory and Its Effect on Social Media Trends",
    "U.S. Voters' Opinions on Trump's Election Win",
    "Trump Win Reactions: What Americans Are Saying Online",
```

```

    "Trump's Win and America's Political Divide",
    "Public Opinion on Trump's Policies Post-Election",
    "Trump's Impact on U.S. Economy: Public Perception",
    "Trump Administration"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }
            all_articles.append(article_data)

# Save the result as a JSON file
with open('finance16_API.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)

```



```
print(f"Total number of articles fetched: {total_articles}")  
print("Articles saved to 'finance16_API.json'")
```

Fetching articles for topic: Trump's Election Impact on U.S. Financial Markets
Error fetching data for topic 'Trump's Election Impact on U.S. Financial Markets': 429
Fetching articles for topic: Trump's Victory and Wall Street Reactions
Error fetching data for topic 'Trump's Victory and Wall Street Reactions': 429
Fetching articles for topic: Post-Trump Election Financial Policies
Error fetching data for topic 'Post-Trump Election Financial Policies': 429
Fetching articles for topic: Trump's Win and U.S. Stock Market Trends
Error fetching data for topic 'Trump's Win and U.S. Stock Market Trends': 429
Fetching articles for topic: Trump's Election and Financial Sector Outlook
Error fetching data for topic 'Trump's Election and Financial Sector Outlook': 429
Fetching articles for topic: Trump's Victory and U.S. Economic Policies
Error fetching data for topic 'Trump's Victory and U.S. Economic Policies': 429
Fetching articles for topic: Trump's Win and U.S. Financial Regulations
Error fetching data for topic 'Trump's Win and U.S. Financial Regulations': 429
Fetching articles for topic: Trump's Election and U.S. Banking Sector
Error fetching data for topic 'Trump's Election and U.S. Banking Sector': 429
Fetching articles for topic: Trump's Victory and U.S. Investment Strategies
Error fetching data for topic 'Trump's Victory and U.S. Investment Strategies': 429
Fetching articles for topic: Trump's Win and U.S. Financial News
Error fetching data for topic 'Trump's Win and U.S. Financial News': 429
Fetching articles for topic: Americans' Reactions to Trump's Win
Error fetching data for topic 'Americans' Reactions to Trump's Win': 429
Fetching articles for topic: Trump's Win and American Public Sentiment
Error fetching data for topic 'Trump's Win and American Public Sentiment': 429
Fetching articles for topic: Social Media Reactions to Trump's Victory
Fetching articles for topic: Trump's Win and U.S. Voter Reactions
Fetching articles for topic: How Social Media Reacted to Trump's Election
Fetching articles for topic: Americans Discussing Trump on Social Media
Fetching articles for topic: Public Opinion on Trump's Victory
Fetching articles for topic: Trump's Election and Social Media Buzz
Error fetching data for topic 'Trump's Election and Social Media Buzz': 429
Fetching articles for topic: Social Media Sentiment on Trump's Win
Error fetching data for topic 'Social Media Sentiment on Trump's Win': 429
Fetching articles for topic: Americans' Feelings After Trump's Victory
Error fetching data for topic 'Americans' Feelings After Trump's Victory': 429
Fetching articles for topic: Trump's Win and American Emotions on Social Media
Error fetching data for topic 'Trump's Win and American Emotions on Social Media': 429
Fetching articles for topic: Trump's Election: What Do Americans Really Think?
Error fetching data for topic 'Trump's Election: What Do Americans Really Think?': 429
Fetching articles for topic: Public Sentiment on Trump's Victory in 2024
Error fetching data for topic 'Public Sentiment on Trump's Victory in 2024': 429
Fetching articles for topic: Social Media's Response to Trump Winning Again
Error fetching data for topic 'Social Media's Response to Trump Winning Again': 429
Fetching articles for topic: Trump's Win and Its Impact on American Society
Error fetching data for topic 'Trump's Win and Its Impact on American Society': 429
Fetching articles for topic: How Trump's Election Affected U.S. Public Opinion
Error fetching data for topic 'How Trump's Election Affected U.S. Public Opinion': 429

Fetching articles for topic: Trump's Victory: Divided Reactions Among Americans
Error fetching data for topic 'Trump's Victory: Divided Reactions Among Americans': 429
Fetching articles for topic: Social Media Discussions on Trump's Second Term
Error fetching data for topic 'Social Media Discussions on Trump's Second Term': 429
Fetching articles for topic: Trump's Win and the Future of U.S. Politics
Error fetching data for topic 'Trump's Win and the Future of U.S. Politics': 429
Fetching articles for topic: Reactions to Trump's Election on Twitter and Facebook
Error fetching data for topic 'Reactions to Trump's Election on Twitter and Facebook': 429
Fetching articles for topic: How Did Trump's Win Affect American Families?
Error fetching data for topic 'How Did Trump's Win Affect American Families?': 429
Fetching articles for topic: Trump's Presidency: Public Views and Media Reaction
Error fetching data for topic 'Trump's Presidency: Public Views and Media Reaction': 429
Fetching articles for topic: American Reactions to Trump's Policies After Winning
Error fetching data for topic 'American Reactions to Trump's Policies After Winning': 429
Fetching articles for topic: Trump's Victory and Its Effect on Social Media Trends
Error fetching data for topic 'Trump's Victory and Its Effect on Social Media Trends': 429
Fetching articles for topic: U.S. Voters' Opinions on Trump's Election Win
Error fetching data for topic 'U.S. Voters' Opinions on Trump's Election Win': 429
Fetching articles for topic: Trump Win Reactions: What Americans Are Saying Online
Error fetching data for topic 'Trump Win Reactions: What Americans Are Saying Online': 429
Fetching articles for topic: Trump's Win and America's Political Divide
Error fetching data for topic 'Trump's Win and America's Political Divide': 429
Fetching articles for topic: Public Opinion on Trump's Policies Post-Election
Error fetching data for topic 'Public Opinion on Trump's Policies Post-Election': 429
Fetching articles for topic: Trump's Impact on U.S. Economy: Public Perception
Error fetching data for topic 'Trump's Impact on U.S. Economy: Public Perception': 429
Fetching articles for topic: Trump Administration
Error fetching data for topic 'Trump Administration': 429
Total number of articles fetched: 50
Articles saved to 'finance16_API.json'


```

In [73]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of topics you're interested in
topics = [
    "U.S. Financial Markets", "U.S.A."
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }

```

```
    }
    all_articles.append(article_data)

# Save the result as a JSON file
with open('finance17_API.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'finance17_API.json'")
```

```
Fetching articles for topic: U.S. Financial Markets
Fetching articles for topic: U.S.A.
Total number of articles fetched: 20
Articles saved to 'finance17_API.json'
```



```

In [70]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of selected topics focused on Trump's return as president and its impact on the economy
topics = [
    "Trump's Election Impact on U.S. Financial Markets",
    "Trump's Victory and Wall Street Reactions",
    "Post-Trump Election Financial Policies",
    "Trump's Win and U.S. Stock Market Trends",
    "Trump's Election and Financial Sector Outlook",
    "Trump's Victory and U.S. Economic Policies",
    "Trump's Win and U.S. Financial Regulations",
    "Trump's Election and U.S. Banking Sector",
    "Trump's Victory and U.S. Investment Strategies",
    "Trump's Win and U.S. Financial News",
    "Trump's Impact on U.S. Economy: Public Perception",
    "Trump Administration"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:

```



```

print(f"Fetching articles for topic: {topic}")
articles = fetch_articles(topic)

if articles:
    for article in articles:
        # Extract necessary data (Title, Date, URL)
        article_data = {
            "title": article.get('headline', {}).get('main', 'No title'),
            "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
            "url": article.get('web_url', 'No URL'),
            "topic": topic
        }
        all_articles.append(article_data)

# Save the result as a JSON file
with open('finance18_API.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'finance18_API.json'")

```

```

Fetching articles for topic: Trump's Election Impact on U.S. Financial Markets
Fetching articles for topic: Trump's Victory and Wall Street Reactions
Fetching articles for topic: Post-Trump Election Financial Policies
Fetching articles for topic: Trump's Win and U.S. Stock Market Trends
Error fetching data for topic 'Trump's Win and U.S. Stock Market Trends': 429
Fetching articles for topic: Trump's Election and Financial Sector Outlook
Error fetching data for topic 'Trump's Election and Financial Sector Outlook': 429
Fetching articles for topic: Trump's Victory and U.S. Economic Policies
Error fetching data for topic 'Trump's Victory and U.S. Economic Policies': 429
Fetching articles for topic: Trump's Win and U.S. Financial Regulations
Error fetching data for topic 'Trump's Win and U.S. Financial Regulations': 429
Fetching articles for topic: Trump's Election and U.S. Banking Sector
Fetching articles for topic: Trump's Victory and U.S. Investment Strategies
Fetching articles for topic: Trump's Win and U.S. Financial News
Fetching articles for topic: Trump's Impact on U.S. Economy: Public Perception
Fetching articles for topic: Trump Administration
Total number of articles fetched: 80
Articles saved to 'finance18_API.json'

```



```

In [69]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS62012ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of selected topics focused on Trump's return as president and its impact on the economy
topics = [
    "Donald Trump", "Economy"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }

```

```
    }
    all_articles.append(article_data)

# Save the result as a JSON file
with open('finance19_API.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'finance19_API.json'")
```

Fetching articles for topic: Donald Trump

Fetching articles for topic: Economy

Total number of articles fetched: 20

Articles saved to 'finance19_API.json'


```

In [1]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics to search for related to Trump's presidency and defense
topics = [
    "Trump's Presidency and U.S. Military Policies",
    "Trump's Defense Strategies and National Security",
    "Trump's Approach to NATO and Defense Alliances",
    "Trump's Military Spending and Budget Plans",
    "Impact of Trump's Policies on U.S. Armed Forces",
    "Trump's Decisions on U.S. Defense Contracts",
    "Trump's Presidency and U.S. Defense Industry",
    "Trump's Policies on Cybersecurity Defense",
    "Trump's Relations with North Korea and Defense",
    "Trump's Approach to Russia and National Security",
    "Trump's Presidency and the U.S. Space Force",
    "Impact of Trump's Defense Policies on Middle East",
    "Trump's Withdrawal from Global Defense Agreements",
    "Trump's Stance on Nuclear Weapons and Proliferation",
    "Trump's Policies on Border Security and Defense",
    "Trump's Defense Relations with China",
    "Trump's Approach to Homeland Security",
    "Trump's Presidency and Veterans' Affairs",
    "Trump's Decisions on U.S. Defense Deployments",
    "Trump's Military Actions in Afghanistan",
    "Trump's Strategy on Terrorism and Defense",
    "Trump's Influence on Global Defense Dynamics",
    "Public Opinion on Trump's Defense Policies",
    "Trump's Presidency and American Defense Innovation",
    "Trump's Relations with Defense Contractors",
    "Trump's Presidency and Global Military Balance",
    "Trump's Approach to Defense Partnerships with Allies",
    "Trump's Presidency and National Defense Funding",
    "Impact of Trump's Policies on Military Technology",
    "Trump's Defense Policies and U.S. Foreign Relations",
    "Trump's Administration and Global Security Challenges"
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

```

```

while len(all_articles) < total_articles:
    params = {
        "q": topic,                # Query keyword
        "apiKey": API_KEY,         # API Key
        "language": "en",          # Language preference
        "sortBy": "relevance",     # Sort articles by relevance
        "pageSize": page_size,     # Number of articles per request
        "page": page               # Pagination page number
    }

    response = requests.get(BASE_URL, params=params)

    if response.status_code == 200:
        articles = response.json()["articles"]
        for article in articles:
            # Only extract needed fields including date
            all_articles.append({
                "title": article.get("title"),
                "description": article.get("description"),
                "url": article.get("url"),
                "publishedAt": article.get("publishedAt"),
                "source": article["source"]["name"]
            })
        print(f"Fetched {len(articles)} articles from page {page}")
        page += 1
    else:
        print(f"Error fetching news for topic '{topic}': {response.status_code}")
        break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("trump_defense_policies_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

```



```
# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")
```

Fetching news for: Trump's Presidency and U.S. Military Policies
Fetched 66 articles from page 1
Error fetching news for topic 'Trump's Presidency and U.S. Military Policies': 426
Total articles fetched for 'Trump's Presidency and U.S. Military Policies': 66

Fetching news for: Trump's Defense Strategies and National Security
Fetched 13 articles from page 1
Error fetching news for topic 'Trump's Defense Strategies and National Security': 426
Total articles fetched for 'Trump's Defense Strategies and National Security': 13

Fetching news for: Trump's Approach to NATO and Defense Alliances
Fetched 3 articles from page 1
Error fetching news for topic 'Trump's Approach to NATO and Defense Alliances': 426
Total articles fetched for 'Trump's Approach to NATO and Defense Alliances': 3

Fetching news for: Trump's Military Spending and Budget Plans
Fetched 83 articles from page 1
Error fetching news for topic 'Trump's Military Spending and Budget Plans': 426
Total articles fetched for 'Trump's Military Spending and Budget Plans': 83

Fetching news for: Impact of Trump's Policies on U.S. Armed Forces
Fetched 8 articles from page 1
Error fetching news for topic 'Impact of Trump's Policies on U.S. Armed Forces': 426
Total articles fetched for 'Impact of Trump's Policies on U.S. Armed Forces': 8

Fetching news for: Trump's Decisions on U.S. Defense Contracts
Fetched 13 articles from page 1
Error fetching news for topic 'Trump's Decisions on U.S. Defense Contracts': 426
Total articles fetched for 'Trump's Decisions on U.S. Defense Contracts': 13

Fetching news for: Trump's Presidency and U.S. Defense Industry
Fetched 31 articles from page 1
Error fetching news for topic 'Trump's Presidency and U.S. Defense Industry': 426
Total articles fetched for 'Trump's Presidency and U.S. Defense Industry': 31

Fetching news for: Trump's Policies on Cybersecurity Defense
Fetched 23 articles from page 1
Error fetching news for topic 'Trump's Policies on Cybersecurity Defense': 426
Total articles fetched for 'Trump's Policies on Cybersecurity Defense': 23

Fetching news for: Trump's Relations with North Korea and Defense
Fetched 22 articles from page 1
Error fetching news for topic 'Trump's Relations with North Korea and Defense': 426
Total articles fetched for 'Trump's Relations with North Korea and Defense': 22

Fetching news for: Trump's Approach to Russia and National Security

Fetches 100 articles from page 1
Error fetching news for topic 'Trump's Approach to Russia and National Security': 426
Total articles fetched for 'Trump's Approach to Russia and National Security': 100

Fetching news for: Trump's Presidency and the U.S. Space Force
Fetched 27 articles from page 1
Error fetching news for topic 'Trump's Presidency and the U.S. Space Force': 426
Total articles fetched for 'Trump's Presidency and the U.S. Space Force': 27

Fetching news for: Impact of Trump's Defense Policies on Middle East
Fetched 12 articles from page 1
Error fetching news for topic 'Impact of Trump's Defense Policies on Middle East': 426
Total articles fetched for 'Impact of Trump's Defense Policies on Middle East': 12

Fetching news for: Trump's Withdrawal from Global Defense Agreements
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Withdrawal from Global Defense Agreements': 426
No articles found for 'Trump's Withdrawal from Global Defense Agreements'.

Fetching news for: Trump's Stance on Nuclear Weapons and Proliferation
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Stance on Nuclear Weapons and Proliferation': 426
No articles found for 'Trump's Stance on Nuclear Weapons and Proliferation'.

Fetching news for: Trump's Policies on Border Security and Defense
Fetched 61 articles from page 1
Error fetching news for topic 'Trump's Policies on Border Security and Defense': 426
Total articles fetched for 'Trump's Policies on Border Security and Defense': 61

Fetching news for: Trump's Defense Relations with China
Fetched 69 articles from page 1
Error fetching news for topic 'Trump's Defense Relations with China': 426
Total articles fetched for 'Trump's Defense Relations with China': 69

Fetching news for: Trump's Approach to Homeland Security
Fetched 54 articles from page 1
Error fetching news for topic 'Trump's Approach to Homeland Security': 426
Total articles fetched for 'Trump's Approach to Homeland Security': 54

Fetching news for: Trump's Presidency and Veterans' Affairs
Fetched 37 articles from page 1
Error fetching news for topic 'Trump's Presidency and Veterans' Affairs': 426
Total articles fetched for 'Trump's Presidency and Veterans' Affairs': 37

Fetching news for: Trump's Decisions on U.S. Defense Deployments
Fetched 1 articles from page 1
Error fetching news for topic 'Trump's Decisions on U.S. Defense Deployments': 426
Total articles fetched for 'Trump's Decisions on U.S. Defense Deployments': 1

Fetching news for: Trump's Military Actions in Afghanistan

Fetches 16 articles from page 1

Error fetching news for topic 'Trump's Military Actions in Afghanistan': 426

Total articles fetched for 'Trump's Military Actions in Afghanistan': 16

Fetching news for: Trump's Strategy on Terrorism and Defense

Fetches 4 articles from page 1

Error fetching news for topic 'Trump's Strategy on Terrorism and Defense': 426

Total articles fetched for 'Trump's Strategy on Terrorism and Defense': 4

Fetching news for: Trump's Influence on Global Defense Dynamics

Fetches 1 articles from page 1

Error fetching news for topic 'Trump's Influence on Global Defense Dynamics': 426

Total articles fetched for 'Trump's Influence on Global Defense Dynamics': 1

Fetching news for: Public Opinion on Trump's Defense Policies

Fetches 34 articles from page 1

Error fetching news for topic 'Public Opinion on Trump's Defense Policies': 426

Total articles fetched for 'Public Opinion on Trump's Defense Policies': 34

Fetching news for: Trump's Presidency and American Defense Innovation

Fetches 6 articles from page 1

Error fetching news for topic 'Trump's Presidency and American Defense Innovation': 426

Total articles fetched for 'Trump's Presidency and American Defense Innovation': 6

Fetching news for: Trump's Relations with Defense Contractors

Fetches 2 articles from page 1

Error fetching news for topic 'Trump's Relations with Defense Contractors': 426

Total articles fetched for 'Trump's Relations with Defense Contractors': 2

Fetching news for: Trump's Presidency and Global Military Balance

Fetches 3 articles from page 1

Error fetching news for topic 'Trump's Presidency and Global Military Balance': 426

Total articles fetched for 'Trump's Presidency and Global Military Balance': 3

Fetching news for: Trump's Approach to Defense Partnerships with Allies

Fetches 2 articles from page 1

Error fetching news for topic 'Trump's Approach to Defense Partnerships with Allies': 426

Total articles fetched for 'Trump's Approach to Defense Partnerships with Allies': 2

Fetching news for: Trump's Presidency and National Defense Funding

Fetches 35 articles from page 1

Error fetching news for topic 'Trump's Presidency and National Defense Funding': 426

Total articles fetched for 'Trump's Presidency and National Defense Funding': 35

Fetching news for: Impact of Trump's Policies on Military Technology

Fetches 5 articles from page 1

Error fetching news for topic 'Impact of Trump's Policies on Military Technology': 426
Total articles fetched for 'Impact of Trump's Policies on Military Technology': 5

Fetching news for: Trump's Defense Policies and U.S. Foreign Relations

Fetched 29 articles from page 1

Error fetching news for topic 'Trump's Defense Policies and U.S. Foreign Relations': 426

Total articles fetched for 'Trump's Defense Policies and U.S. Foreign Relations': 29

Fetching news for: Trump's Administration and Global Security Challenges

Fetched 52 articles from page 1

Error fetching news for topic 'Trump's Administration and Global Security Challenges': 426

Total articles fetched for 'Trump's Administration and Global Security Challenges': 52

Total number of articles fetched: 812


```
In [24]: import json

# Path to the input file
input_file = "trump_defense_policies_with_dates.json"
output_file = "defence8_with_dates_API_Reddit.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Check if `data` is a dictionary
if isinstance(data, dict):
    topics_data = data
else:
    # If `data` is a List, use the first item (if that's the actual structure)
    topics_data = data[0]

# Initialize a List to store processed articles
processed_articles = []

# Iterate through the topics and their articles
for topic, articles in topics_data.items():
    for article in articles:
        # Handle `source` field
        source_name = None
        if isinstance(article.get("source"), dict):
            source_name = article["source"].get("name")
        elif isinstance(article.get("source"), str):
            source_name = article["source"]

        processed_article = {
            "topic": topic,
            "source_name": source_name,
            "author": article.get("author"),
            "title": article.get("title"),
            "description": article.get("description"),
            "url": article.get("url"),
            "urlToImage": article.get("urlToImage"),
            "date": article.get("publishedAt"), # Changed key from `publishedAt` to `date`
            "content": article.get("content"),
        }
        # Append the processed article to the List
        processed_articles.append(processed_article)

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(processed_articles, file, indent=4)
```

```
print(f"Cleaned and structured data saved to {output_file}")
```

Cleaned and structured data saved to defence8_with_dates_API_Reddit.json


```

In [2]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics to search for related to Trump's presidency and defense
topics = [
    "Trump's Defense Policies on China and Taiwan",
    "Trump's Withdrawal of U.S. Troops from Syria",
    "Trump's Policies on Defense Spending Cuts",
    "Trump's Military Support to Israel",
    "Trump's Decisions on NATO Funding",
    "Trump's Cybersecurity Initiatives in National Defense",
    "Trump's Military Exercises in the South China Sea",
    "Trump's Defense Approach to Russian Aggression",
    "Trump's Space Force and U.S. Space Defense Policies",
    "Trump's Impact on U.S. Global Defense Commitments",
    "Trump's Strategies on Counterterrorism Operations",
    "Trump's Modernization of the U.S. Nuclear Arsenal",
    "Trump's Role in Promoting Cyber Defense Technologies",
    "Trump's Policies on Hypersonic Weapons Development",
    "Trump's Presidency and Artificial Intelligence in Defense",
    "Trump's Military Aid to Ukraine",
    "Trump's Approach to Defense Relations with European Allies",
    "Trump's Military Strategy in the Indo-Pacific Region",
    "Trump's Relations with NATO and U.S. Commitments",
    "Trump's Border Defense and Homeland Security Policies"
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

    while len(all_articles) < total_articles:
        params = {
            "q": topic, # Query keyword
            "apiKey": API_KEY, # API Key
            "language": "en", # Language preference
            "sortBy": "relevance", # Sort articles by relevance
            "pageSize": page_size, # Number of articles per request
            "page": page # Pagination page number
        }

```

```

response = requests.get(BASE_URL, params=params)

if response.status_code == 200:
    articles = response.json()["articles"]
    for article in articles:
        # Only extract needed fields including date
        all_articles.append({
            "title": article.get("title"),
            "description": article.get("description"),
            "url": article.get("url"),
            "publishedAt": article.get("publishedAt"),
            "source": article["source"]["name"]
        })
    print(f"Fetched {len(articles)} articles from page {page}")
    page += 1
else:
    print(f"Error fetching news for topic '{topic}': {response.status_code}")
    break

return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("trump_defense_policies2_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")

```

Fetching news for: Trump's Defense Policies on China and Taiwan
Fetched 10 articles from page 1
Error fetching news for topic 'Trump's Defense Policies on China and Taiwan': 426
Total articles fetched for 'Trump's Defense Policies on China and Taiwan': 10

Fetching news for: Trump's Withdrawal of U.S. Troops from Syria
Fetched 4 articles from page 1
Error fetching news for topic 'Trump's Withdrawal of U.S. Troops from Syria': 426
Total articles fetched for 'Trump's Withdrawal of U.S. Troops from Syria': 4

Fetching news for: Trump's Policies on Defense Spending Cuts
Fetched 47 articles from page 1
Error fetching news for topic 'Trump's Policies on Defense Spending Cuts': 426
Total articles fetched for 'Trump's Policies on Defense Spending Cuts': 47

Fetching news for: Trump's Military Support to Israel
Fetched 100 articles from page 1
Error fetching news for topic 'Trump's Military Support to Israel': 426
Total articles fetched for 'Trump's Military Support to Israel': 100

Fetching news for: Trump's Decisions on NATO Funding
Fetched 26 articles from page 1
Error fetching news for topic 'Trump's Decisions on NATO Funding': 426
Total articles fetched for 'Trump's Decisions on NATO Funding': 26

Fetching news for: Trump's Cybersecurity Initiatives in National Defense
Fetched 2 articles from page 1
Error fetching news for topic 'Trump's Cybersecurity Initiatives in National Defense': 426
Total articles fetched for 'Trump's Cybersecurity Initiatives in National Defense': 2

Fetching news for: Trump's Military Exercises in the South China Sea
Fetched 5 articles from page 1
Error fetching news for topic 'Trump's Military Exercises in the South China Sea': 426
Total articles fetched for 'Trump's Military Exercises in the South China Sea': 5

Fetching news for: Trump's Defense Approach to Russian Aggression
Fetched 3 articles from page 1
Error fetching news for topic 'Trump's Defense Approach to Russian Aggression': 426
Total articles fetched for 'Trump's Defense Approach to Russian Aggression': 3

Fetching news for: Trump's Space Force and U.S. Space Defense Policies
Fetched 20 articles from page 1
Error fetching news for topic 'Trump's Space Force and U.S. Space Defense Policies': 426
Total articles fetched for 'Trump's Space Force and U.S. Space Defense Policies': 20

Fetching news for: Trump's Impact on U.S. Global Defense Commitments

Fetched 2 articles from page 1
 Error fetching news for topic 'Trump's Impact on U.S. Global Defense Commitments': 426
 Total articles fetched for 'Trump's Impact on U.S. Global Defense Commitments': 2

Fetching news for: Trump's Strategies on Counterterrorism Operations
 Fetched 0 articles from page 1
 Error fetching news for topic 'Trump's Strategies on Counterterrorism Operations': 426
 No articles found for 'Trump's Strategies on Counterterrorism Operations'.

Fetching news for: Trump's Modernization of the U.S. Nuclear Arsenal
 Fetched 0 articles from page 1
 Error fetching news for topic 'Trump's Modernization of the U.S. Nuclear Arsenal': 426
 No articles found for 'Trump's Modernization of the U.S. Nuclear Arsenal'.

Fetching news for: Trump's Role in Promoting Cyber Defense Technologies
 Fetched 0 articles from page 1
 Error fetching news for topic 'Trump's Role in Promoting Cyber Defense Technologies': 426
 No articles found for 'Trump's Role in Promoting Cyber Defense Technologies'.

Fetching news for: Trump's Policies on Hypersonic Weapons Development
 Fetched 3 articles from page 1
 Error fetching news for topic 'Trump's Policies on Hypersonic Weapons Development': 426
 Total articles fetched for 'Trump's Policies on Hypersonic Weapons Development': 3

Fetching news for: Trump's Presidency and Artificial Intelligence in Defense
 Fetched 27 articles from page 1
 Error fetching news for topic 'Trump's Presidency and Artificial Intelligence in Defense': 426
 Total articles fetched for 'Trump's Presidency and Artificial Intelligence in Defense': 27

Fetching news for: Trump's Military Aid to Ukraine
 Fetched 100 articles from page 1
 Error fetching news for topic 'Trump's Military Aid to Ukraine': 426
 Total articles fetched for 'Trump's Military Aid to Ukraine': 100

Fetching news for: Trump's Approach to Defense Relations with European Allies
 Fetched 18 articles from page 1
 Error fetching news for topic 'Trump's Approach to Defense Relations with European Allies': 426
 Total articles fetched for 'Trump's Approach to Defense Relations with European Allies': 18

Fetching news for: Trump's Military Strategy in the Indo-Pacific Region
 Fetched 5 articles from page 1
 Error fetching news for topic 'Trump's Military Strategy in the Indo-Pacific Region': 426
 Total articles fetched for 'Trump's Military Strategy in the Indo-Pacific Region': 5

Fetching news for: Trump's Relations with NATO and U.S. Commitments
 Fetched 4 articles from page 1
 Error fetching news for topic 'Trump's Relations with NATO and U.S. Commitments': 426
 Total articles fetched for 'Trump's Relations with NATO and U.S. Commitments': 4

```
Fetching news for: Trump's Border Defense and Homeland Security Policies
Error fetching news for topic 'Trump's Border Defense and Homeland Security Policies': 429
No articles found for 'Trump's Border Defense and Homeland Security Policies'.
```

```
Total number of articles fetched: 376
```

In [25]:

```
Cleaned and structured data saved to defence9_with_dates_API_Reddit.json
```

```
....
```



```
In [3]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics related to Donald Trump, defense industry, and public sentiment
topics = [
    # Sentiments and Public Opinion
    "Public Reactions to Trump's Defense Policies",
    "American Views on Trump's Military Spending",
    "Sentiments on Trump's Withdrawal of Troops from Syria",
    "How Americans View Trump's Support for NATO",
    "Public Opinion on Trump's Space Force Initiative",
    "Reactions to Trump's Cybersecurity Defense Strategies",
    "Social Media Discussions on Trump's Defense Budget Increases",
    "Public Reactions to Trump's Military Aid to Ukraine",
    "How Americans Feel About Trump's Approach to North Korea",
    "Reactions to Trump's Defense Relations with Saudi Arabia",

    # Impact on the Defense Industry
    "Impact of Trump's Policies on U.S. Defense Contractors",
    "How Trump's Defense Spending Affected Military Jobs",
    "Reactions to Trump's Role in Hypersonic Weapons Development",
    "Trump's Policies and Their Effect on Defense Technology Innovation",
    "Public Opinion on Trump's Drone Warfare Policies",
    "How Trump's Presidency Boosted U.S. Defense Manufacturing",
    "Reactions to Trump's Modernization of Nuclear Weapons",
    "Trump's Policies on AI and Their Impact on U.S. Defense",
    "How Trump's Space Force Changed the Defense Industry",
    "Trump's Defense Policies and the Role of Private Companies",

    # Global and Domestic Defense Strategy
    "Reactions to Trump's Policies on NATO Defense Contributions",
    "Public Sentiment on Trump's Stance Against Chinese Aggression",
    "How Americans Viewed Trump's Policies on the Middle East",
    "Trump's Withdrawal from the Iran Nuclear Deal and Its Impacts",
    "Public Opinion on Trump's Defense Strategy for the Korean Peninsula",
    "Reactions to Trump's Border Wall as a National Security Measure",
    "Impact of Trump's Policies on U.S. Cybersecurity Defense",
    "Trump's Relations with Russia and Defense Policy Sentiments",
    "Social Media Response to Trump's Troop Deployments Abroad",
    "Public Reactions to Trump's National Guard Deployments",

    # Influence on Veterans and Society
    "How Trump's Policies Affected U.S. Veterans",
```



```
"Public Sentiment on Trump's Support for Military Families",  
"Reactions to Trump's Handling of Military Protests",  
"Social Media Sentiments on Trump's Policies for Veterans' Healthcare",  
"Trump's Military Honors and Public Perception",
```

Historical and Long-term Impact

```
"Americans Reflect on Trump's Defense Policies Post-Presidency",  
"Legacy of Trump's Military Policies in U.S. History",  
"Long-term Impact of Trump's Defense Spending on U.S. Economy",  
"Trump's Defense Industry Legacy and Public Perception",  
"How Trump's Defense Policies Shaped America's Global Role"
```

```
]
```

Function to fetch news articles for a topic with pagination

```
def fetch_news(topic, total_articles=2000):  
    all_articles = []  
    page = 1  
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)  
  
    while len(all_articles) < total_articles:  
        params = {  
            "q": topic, # Query keyword  
            "apiKey": API_KEY, # API Key  
            "language": "en", # Language preference  
            "sortBy": "relevance", # Sort articles by relevance  
            "pageSize": page_size, # Number of articles per request  
            "page": page # Pagination page number  
        }  
  
        response = requests.get(BASE_URL, params=params)  
  
        if response.status_code == 200:  
            articles = response.json()["articles"]  
            for article in articles:  
                # Only extract needed fields including date  
                all_articles.append({  
                    "title": article.get("title"),  
                    "description": article.get("description"),  
                    "url": article.get("url"),  
                    "publishedAt": article.get("publishedAt"),  
                    "source": article["source"]["name"]  
                })  
            print(f"Fetchd {len(articles)} articles from page {page}")  
            page += 1  
        else:  
            print(f"Error fetching news for topic '{topic}': {response.status_code}")  
            break
```

```

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("trump_defense_policies3_with_dates.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")

```

```

Fetching news for: Public Reactions to Trump's Defense Policies
Error fetching news for topic 'Public Reactions to Trump's Defense Policies': 429
No articles found for 'Public Reactions to Trump's Defense Policies'.

Fetching news for: American Views on Trump's Military Spending
Error fetching news for topic 'American Views on Trump's Military Spending': 429
No articles found for 'American Views on Trump's Military Spending'.

Fetching news for: Sentiments on Trump's Withdrawal of Troops from Syria
Error fetching news for topic 'Sentiments on Trump's Withdrawal of Troops from Syria': 429
No articles found for 'Sentiments on Trump's Withdrawal of Troops from Syria'.

Fetching news for: How Americans View Trump's Support for NATO
Error fetching news for topic 'How Americans View Trump's Support for NATO': 429
No articles found for 'How Americans View Trump's Support for NATO'.

Fetching news for: Public Opinion on Trump's Space Force Initiative
Error fetching news for topic 'Public Opinion on Trump's Space Force Initiative': 429
No articles found for 'Public Opinion on Trump's Space Force Initiative'.

```

In [13]: !pip install requests

```
Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\admin\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests) (2024.8.30)
```



```
In [13]: import requests
import json
from datetime import datetime

# Your GNews API key
api_key = "1bd88a87e8c400204840189f8484382e"

# Base URL for GNews API
base_url = "https://gnews.io/api/v4/search"

topics = [
    "Trump defense policies",
    "Trump military spending",
    "Trump national security",
    "Trump defense budget",
    "Trump Pentagon",
    "Trump NATO policies",
    "Trump arms deals",
    "Trump defense contracts",
    "Trump and US military",
    "Trump foreign policy and defense",
    "Trump veterans policies",
    "Trump military alliances",
    "Trump nuclear policies",
    "Trump missile defense",
    "Trump military deployments",
    "Trump and defense industry",
    "Trump and military reforms",
    "Trump defense priorities",
    "Trump and military bases",
    "Trump cybersecurity defense"
]

# Function to fetch news articles for a specific topic with pagination
def fetch_news(topic, page):
    params = {
        'q': topic,
        'token': api_key,
        'lang': 'en', # Search in English
        'sort': 'published_at', # Sort by publication date
        'max': 100, # Set the max number of articles per request
        'page': page # Page number for pagination
    }

    response = requests.get(base_url, params=params)
```

```

# If the request was successful
if response.status_code == 200:
    return response.json()
else:
    print(f"Error fetching news for {topic}. Status Code: {response.status_code}")
    return None

# Function to get all news articles for a specific topic
def get_all_articles_for_topic(topic):
    all_articles = []
    page = 1

    while True:
        news_data = fetch_news(topic, page)

        if news_data and news_data['articles']:
            all_articles.extend(news_data['articles'])
            page += 1 # Move to the next page
        else:
            break # No more articles to fetch

    return all_articles

# Fetch news for each topic and save results in a list
all_articles = []

for topic in topics:
    print(f"Fetching all news articles for topic: {topic}")
    articles = get_all_articles_for_topic(topic)
    # Add timestamp and publication date to each article
    for article in articles:
        article['fetched_at'] = datetime.now().strftime('%Y-%m-%d %H:%M:%S') # Current fetch time
        article['published_at'] = article.get('publishedAt') # Extract published date from the API response
    all_articles.extend(articles)

# Save the collected news data into a JSON file
with open('trump_defense_policies4_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

print("All news articles with dates saved successfully!")

```

Fetching all news articles for topic: Trump defense policies
Error fetching news for Trump defense policies. Status Code: 429
Fetching all news articles for topic: Trump military spending
Error fetching news for Trump military spending. Status Code: 429
Fetching all news articles for topic: Trump national security
Error fetching news for Trump national security. Status Code: 429
Fetching all news articles for topic: Trump defense budget
Error fetching news for Trump defense budget. Status Code: 429
Fetching all news articles for topic: Trump Pentagon
Error fetching news for Trump Pentagon. Status Code: 429
Fetching all news articles for topic: Trump NATO policies
Error fetching news for Trump NATO policies. Status Code: 429
Fetching all news articles for topic: Trump arms deals
Error fetching news for Trump arms deals. Status Code: 429
Fetching all news articles for topic: Trump defense contracts
Error fetching news for Trump defense contracts. Status Code: 429
Fetching all news articles for topic: Trump and US military
Error fetching news for Trump and US military. Status Code: 429
Fetching all news articles for topic: Trump foreign policy and defense
Error fetching news for Trump foreign policy and defense. Status Code: 429
Fetching all news articles for topic: Trump veterans policies
Fetching all news articles for topic: Trump military alliances
Error fetching news for Trump military alliances. Status Code: 429
Fetching all news articles for topic: Trump nuclear policies
Error fetching news for Trump nuclear policies. Status Code: 429
Fetching all news articles for topic: Trump missile defense
Error fetching news for Trump missile defense. Status Code: 403
Fetching all news articles for topic: Trump military deployments
Error fetching news for Trump military deployments. Status Code: 403
Fetching all news articles for topic: Trump and defense industry
Error fetching news for Trump and defense industry. Status Code: 403
Fetching all news articles for topic: Trump and military reforms
Error fetching news for Trump and military reforms. Status Code: 403
Fetching all news articles for topic: Trump defense priorities
Error fetching news for Trump defense priorities. Status Code: 403
Fetching all news articles for topic: Trump and military bases
Error fetching news for Trump and military bases. Status Code: 403
Fetching all news articles for topic: Trump cybersecurity defense
Error fetching news for Trump cybersecurity defense. Status Code: 403
All news articles with dates saved successfully!


```
In [18]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of topics you're interested in
topics = [
    "Trump defense policies",
    "Trump military spending",
    "Trump national security",
    "Trump defense budget",
    "Trump Pentagon",
    "Trump NATO policies",
    "Trump arms deals",
    "Trump defense contracts",
    "Trump and US military",
    "Trump foreign policy and defense",
    "Trump veterans policies",
    "Trump military alliances",
    "Trump nuclear policies",
    "Trump missile defense",
    "Trump military deployments",
    "Trump and defense industry",
    "Trump and military reforms",
    "Trump defense priorities",
    "Trump and military bases",
    "Trump cybersecurity defense"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
```

```

    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }
            all_articles.append(article_data)

# Save the result as a JSON file
with open('trump_defense_policies5_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'trump_defense_policies5_with_dates.json'")

```

Fetching articles for topic: Trump defense policies
Error fetching data for topic 'Trump defense policies': 429
Fetching articles for topic: Trump military spending
Error fetching data for topic 'Trump military spending': 429
Fetching articles for topic: Trump national security
Error fetching data for topic 'Trump national security': 429
Fetching articles for topic: Trump defense budget
Error fetching data for topic 'Trump defense budget': 429
Fetching articles for topic: Trump Pentagon
Error fetching data for topic 'Trump Pentagon': 429
Fetching articles for topic: Trump NATO policies
Error fetching data for topic 'Trump NATO policies': 429
Fetching articles for topic: Trump arms deals
Error fetching data for topic 'Trump arms deals': 429
Fetching articles for topic: Trump defense contracts
Error fetching data for topic 'Trump defense contracts': 429
Fetching articles for topic: Trump and US military
Error fetching data for topic 'Trump and US military': 429
Fetching articles for topic: Trump foreign policy and defense
Error fetching data for topic 'Trump foreign policy and defense': 429
Fetching articles for topic: Trump veterans policies
Error fetching data for topic 'Trump veterans policies': 429
Fetching articles for topic: Trump military alliances
Error fetching data for topic 'Trump military alliances': 429
Fetching articles for topic: Trump nuclear policies
Error fetching data for topic 'Trump nuclear policies': 429
Fetching articles for topic: Trump missile defense
Fetching articles for topic: Trump military deployments
Fetching articles for topic: Trump and defense industry
Fetching articles for topic: Trump and military reforms
Fetching articles for topic: Trump defense priorities
Fetching articles for topic: Trump and military bases
Error fetching data for topic 'Trump and military bases': 429
Fetching articles for topic: Trump cybersecurity defense
Error fetching data for topic 'Trump cybersecurity defense': 429
Total number of articles fetched: 50
Articles saved to 'trump_defense_policies5_with_dates.json'


```

In [17]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of topics you're interested in
topics = [
    "U.S. Defence", "U.S.A.", "Again Trump", "U.S. Military"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }

```

```

    }
    all_articles.append(article_data)

# Save the result as a JSON file
with open('trump_defense_policies6_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'trump_defense_policies6_with_dates.json'")

```

```

Fetching articles for topic: U.S. Defence
Error fetching data for topic 'U.S. Defence': 429
Fetching articles for topic: U.S.A.
Error fetching data for topic 'U.S.A.': 429
Fetching articles for topic: Again Trump
Error fetching data for topic 'Again Trump': 429
Fetching articles for topic: U.S. Military
Error fetching data for topic 'U.S. Military': 429
Total number of articles fetched: 0
Articles saved to 'trump_defense_policies6_with_dates.json'

```



```
In [19]: import requests
import json
from datetime import datetime

# Set your API credentials
app_id = "25ac27f0-17c4-47b9-b6cb-342ee07d1336"
api_key = "XKPvS620l2ujylIkB2zNFXbAhPce2oKb"
secret_key = "a3gfFQnCfEuadFZG"

# List of selected topics focused on Trump's return as president and its impact on the economy
topics = [
    "Trump's Defense Policies and U.S. Security",
    "Trump's Military Spending and Its Impact on U.S. Defense",
    "Trump's Influence on U.S. National Security Strategy",
    "Trump's Stance on NATO and U.S. Global Defense Alliances",
    "Trump's Military Budget and U.S. Armed Forces",
    "Trump's Impact on U.S. Defense Industry",
    "Trump's Foreign Military Engagements",
    "Trump's Defense Reforms in the U.S.",
    "Trump's Policies on U.S. Cybersecurity Defense",
    "Trump and the U.S. Military Strategy",
    "Trump's Military Contracting and Defense Procurement",
    "Trump's Approach to U.S. Missile Defense Systems",
    "Trump's Impact on U.S. Veterans Affairs and Military Support",
    "Trump's Role in Shaping U.S. Defense Posture",
    "Trump's Approach to U.S. Military Bases Abroad",
    "Trump's Views on U.S. Nuclear Defense and Arms Control",
    "Trump's Policy on U.S. Defense Spending Cuts",
    "Trump and the U.S. Space Force",
    "Trump's Impact on U.S. Defense Technology",
    "Trump's Influence on U.S. Military Training and Readiness"
]

# Function to fetch articles for a given topic
def fetch_articles(topic):
    url = f"https://api.nytimes.com/svc/search/v2/articlesearch.json"
    params = {
        'q': topic,
        'api-key': api_key,
    }

    response = requests.get(url, params=params)

    if response.status_code == 200:
        articles = response.json().get('response', {}).get('docs', [])
        return articles
```



```

    else:
        print(f"Error fetching data for topic '{topic}': {response.status_code}")
        return []

# Store all articles in a list
all_articles = []

# Loop through all the topics and fetch articles
for topic in topics:
    print(f"Fetching articles for topic: {topic}")
    articles = fetch_articles(topic)

    if articles:
        for article in articles:
            # Extract necessary data (Title, Date, URL)
            article_data = {
                "title": article.get('headline', {}).get('main', 'No title'),
                "date": article.get('pub_date', 'No date'), # Renamed 'pub_date' to 'date'
                "url": article.get('web_url', 'No URL'),
                "topic": topic
            }
            all_articles.append(article_data)

# Save the result as a JSON file
with open('trump_defense_policies6_with_dates.json', 'w') as json_file:
    json.dump(all_articles, json_file, indent=4)

# Output total number of articles fetched
total_articles = len(all_articles)
print(f"Total number of articles fetched: {total_articles}")
print("Articles saved to 'trump_defense_policies6_with_dates.json'")

```

Fetching articles for topic: Trump's Defense Policies and U.S. Security
Fetching articles for topic: Trump's Military Spending and Its Impact on U.S. Defense
Fetching articles for topic: Trump's Influence on U.S. National Security Strategy
Fetching articles for topic: Trump's Stance on NATO and U.S. Global Defense Alliances
Fetching articles for topic: Trump's Military Budget and U.S. Armed Forces
Fetching articles for topic: Trump's Impact on U.S. Defense Industry
Error fetching data for topic 'Trump's Impact on U.S. Defense Industry': 429
Fetching articles for topic: Trump's Foreign Military Engagements
Error fetching data for topic 'Trump's Foreign Military Engagements': 429
Fetching articles for topic: Trump's Defense Reforms in the U.S.
Error fetching data for topic 'Trump's Defense Reforms in the U.S.': 429
Fetching articles for topic: Trump's Policies on U.S. Cybersecurity Defense
Error fetching data for topic 'Trump's Policies on U.S. Cybersecurity Defense': 429
Fetching articles for topic: Trump and the U.S. Military Strategy
Error fetching data for topic 'Trump and the U.S. Military Strategy': 429
Fetching articles for topic: Trump's Military Contracting and Defense Procurement
Error fetching data for topic 'Trump's Military Contracting and Defense Procurement': 429
Fetching articles for topic: Trump's Approach to U.S. Missile Defense Systems
Error fetching data for topic 'Trump's Approach to U.S. Missile Defense Systems': 429
Fetching articles for topic: Trump's Impact on U.S. Veterans Affairs and Military Support
Error fetching data for topic 'Trump's Impact on U.S. Veterans Affairs and Military Support': 429
Fetching articles for topic: Trump's Role in Shaping U.S. Defense Posture
Error fetching data for topic 'Trump's Role in Shaping U.S. Defense Posture': 429
Fetching articles for topic: Trump's Approach to U.S. Military Bases Abroad
Error fetching data for topic 'Trump's Approach to U.S. Military Bases Abroad': 429
Fetching articles for topic: Trump's Views on U.S. Nuclear Defense and Arms Control
Error fetching data for topic 'Trump's Views on U.S. Nuclear Defense and Arms Control': 429
Fetching articles for topic: Trump's Policy on U.S. Defense Spending Cuts
Error fetching data for topic 'Trump's Policy on U.S. Defense Spending Cuts': 429
Fetching articles for topic: Trump and the U.S. Space Force
Error fetching data for topic 'Trump and the U.S. Space Force': 429
Fetching articles for topic: Trump's Impact on U.S. Defense Technology
Error fetching data for topic 'Trump's Impact on U.S. Defense Technology': 429
Fetching articles for topic: Trump's Influence on U.S. Military Training and Readiness
Error fetching data for topic 'Trump's Influence on U.S. Military Training and Readiness': 429
Total number of articles fetched: 50
Articles saved to 'trump_defense_policies6_with_dates.json'


```

In [33]: import json

# Path to the input file
input_file = "trump_defense_policies_with_dates.json"
output_file = "defence10_with_dates_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Check if `data` is a dictionary
if isinstance(data, dict):
    topics_data = data
else:
    # If `data` is a list, use the first item (if that's the actual structure)
    topics_data = data[0]

# Initialize a list to store processed articles
processed_articles = []

# Iterate through the topics and their articles
for topic, articles in topics_data.items():
    for article in articles:
        # Handle `source` field
        source_name = None
        source = article.get("source")

        if isinstance(source, dict):
            source_name = source.get("name")
        elif isinstance(source, str):
            source_name = source

        processed_article = {
            "topic": topic,
            "source_name": source_name,
            "author": article.get("author"),
            "title": article.get("title"),
            "description": article.get("description"),
            "url": article.get("url"),
            "urlToImage": article.get("urlToImage"),
            "date": article.get("publishedAt"), # Changed key from `publishedAt` to `date`
            "content": article.get("content"),
        }
        # Append the processed article to the list
        processed_articles.append(processed_article)

# Save the cleaned data to a new JSON file

```

```
with open(output_file, "w") as file:  
    json.dump(processed_articles, file, indent=4)  
  
print(f"Cleaned and structured data saved to {output_file}")
```

Cleaned and structured data saved to defence10_with_dates_API.json

```

In [34]: import json
          from datetime import datetime

# Path to the input and output files
input_file = "trump_defense_policies4_with_dates.json"
output_file = "defence11_with_dates_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Initialize a list to store cleaned articles
cleaned_articles = []

# Iterate through the articles and clean the data
for article in data:
    # Clean the publishedAt field to retain only the date
    published_date = article.get("publishedAt", "")
    if "T" in published_date:
        # Convert to date only (YYYY-MM-DD)
        date_obj = datetime.fromisoformat(published_date[:-1]) # Removing the 'Z' and converting to date
        article["publishedAt"] = date_obj.date().isoformat()

    # Create a cleaned article entry
    cleaned_article = {
        "title": article.get("title"),
        "description": article.get("description"),
        "content": article.get("content"),
        "url": article.get("url"),
        "image": article.get("image"),
        "publishedAt": article.get("publishedAt"), # Cleaned date
        "source": article.get("source", {}).get("name"), # Extracting source name only
        "fetched_at": article.get("fetched_at"), # Keeping fetched_at as is
    }

    # Append the cleaned article to the list
    cleaned_articles.append(cleaned_article)

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(cleaned_articles, file, indent=4)

print(f"Cleaned and structured data saved to {output_file}")

```

Cleaned and structured data saved to defence11_with_dates_API.json

```

In [35]: import json
from datetime import datetime

# Path to the input and output files
input_file = "trump_defense_policies5_with_dates.json"
output_file = "defence12_with_dates_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Initialize a list to store cleaned articles
cleaned_articles = []

# Iterate through the articles and clean the data
for article in data:
    # Clean the publishedAt field to retain only the date
    published_date = article.get("publishedAt", "")
    if "T" in published_date:
        # Convert to date only (YYYY-MM-DD)
        date_obj = datetime.fromisoformat(published_date[:-1]) # Removing the 'Z' and converting to date
        article["publishedAt"] = date_obj.date().isoformat()

    # Create a cleaned article entry
    cleaned_article = {
        "title": article.get("title"),
        "description": article.get("description"),
        "content": article.get("content"),
        "url": article.get("url"),
        "image": article.get("image"),
        "publishedAt": article.get("publishedAt"), # Cleaned date
        "source": article.get("source", {}).get("name"), # Extracting source name only
        "fetched_at": article.get("fetched_at"), # Keeping fetched_at as is
    }

    # Append the cleaned article to the list
    cleaned_articles.append(cleaned_article)

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(cleaned_articles, file, indent=4)

print(f"Cleaned and structured data saved to {output_file}")

```

Cleaned and structured data saved to defence12_with_dates_API.json

```

In [36]: import json
from datetime import datetime

# Path to the input and output files
input_file = "trump_defense_policies6_with_dates.json"
output_file = "defence13_with_dates_API.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Initialize a list to store cleaned articles
cleaned_articles = []

# Iterate through the articles and clean the data
for article in data:
    # Clean the publishedAt field to retain only the date
    published_date = article.get("publishedAt", "")
    if "T" in published_date:
        # Convert to date only (YYYY-MM-DD)
        date_obj = datetime.fromisoformat(published_date[:-1]) # Removing the 'Z' and converting to date
        article["publishedAt"] = date_obj.date().isoformat()

    # Create a cleaned article entry
    cleaned_article = {
        "title": article.get("title"),
        "description": article.get("description"),
        "content": article.get("content"),
        "url": article.get("url"),
        "image": article.get("image"),
        "publishedAt": article.get("publishedAt"), # Cleaned date
        "source": article.get("source", {}).get("name"), # Extracting source name only
        "fetched_at": article.get("fetched_at"), # Keeping fetched_at as is
    }

    # Append the cleaned article to the list
    cleaned_articles.append(cleaned_article)

# Save the cleaned data to a new JSON file
with open(output_file, "w") as file:
    json.dump(cleaned_articles, file, indent=4)

print(f"Cleaned and structured data saved to {output_file}")

```

Cleaned and structured data saved to defence13_with_dates_API.json

In [37]: `import json`

```
# Path to the input and output files
input_file = "cleaned_defence_API_Reddit_with_dates.json"
output_file = "2cleaned_defence_API_Reddit_with_dates.json"

# Load the JSON data
with open(input_file, "r") as file:
    data = json.load(file)

# Iterate through the entries and standardize the date field
for entry in data:
    if "date" in entry and entry["date"]:
        # Extract only the date portion (first 10 characters of the string)
        entry["date"] = entry["date"][:10]

# Save the updated JSON data to a new file
with open(output_file, "w") as file:
    json.dump(data, file, indent=4)

print(f"Dates standardized and saved to {output_file}")
```

Dates standardized and saved to 2cleaned_defence_API_Reddit_with_dates.json


```
In [18]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics to search for related to Trump's presidency and defense
topics = [
    "Trump's 2024 Healthcare Plan: Impacts on Affordable Care Act",
    "Reforming U.S. Healthcare: Trump's Strategy for the Next Term",
    "Trump's Proposed Cuts to Medicare and Medicaid: Consequences for Low-Income Americans",
    "Trump's Impact on Healthcare Industry Regulations in 2024",
    "Trump and Prescription Drug Price Reform in the U.S.",
    "Healthcare Access Under Trump: Rural vs. Urban Areas",
    "Trump's Approach to Mental Health Services and Funding",
    "How Trump's 2024 Policies Will Affect U.S. Health Insurance Markets",
    "Trump's Healthcare Vision: Balancing Private vs. Public Healthcare Systems",
    "Trump Administration and Health Equity: Prospects for Minority Communities",
    "The Future of U.S. Healthcare System Under Trump's Second Term",
    "Trump's Stance on Reproductive Health and Women's Rights in Healthcare",
    "Trump's Influence on Public Health Funding and Emergency Response Systems",
    "Trump's Effect on the Pharmaceutical Industry: Policy Shifts and Outcomes",
    "Healthcare Workforce in the Trump Administration: A Changing Landscape",
    "Trump's Efforts to Combat Healthcare Fraud and Abuse in 2024",
    "How Trump's Healthcare Policies Will Shape U.S. Healthcare Innovations",
    "Trump's Impact on Healthcare Accessibility for Veterans in 2024",
    "Will Trump's Healthcare Policies Strengthen or Weaken the Healthcare System?",
    "Trump's 2024 Budget Proposals for U.S. Healthcare System: What's at Stake?",
    "Trump Healthcare Policies 2024",
    "Trump's Impact on U.S. Healthcare System",
    "Trump's Healthcare Reform Plans 2024",
    "Trump's Medicare Changes 2024",
    "Trump's Health Insurance Policies",
    "Trump and U.S. Healthcare Spending",
    "Trump's Healthcare Agenda 2024",
    "Trump and Health Insurance Access",
    "Trump's 2024 Healthcare Budget",
    "Trump's Plan for Prescription Drug Prices",
    "Trump and Affordable Care Act 2024",
    "Trump's Impact on Medicare and Medicaid",
    "Trump's Healthcare Policy Changes",
    "Trump's Views on Healthcare in 2024",
    "Trump and Public Health in 2024",
    "Trump's Plans for U.S. Healthcare",
    "Trump and Healthcare for the Poor",
    "Trump's Healthcare Policy for Seniors",
```

"Trump and Healthcare for Veterans",
 "Trump's Healthcare Reforms",
 "Trump's Healthcare Policy in 2024",
 "Trump's Approach to Health Insurance Reform",
 "Trump and National Healthcare Policies",
 "Trump's Prescription Drug Pricing Plan 2024",
 "Trump's Impact on Healthcare Access 2024",
 "Trump and Health Savings Accounts (HSAs)",
 "Trump's Vision for U.S. Healthcare 2024",
 "Trump and the Affordable Care Act in 2024",
 "Trump's Medicare for All Stance in 2024",
 "Trump's Influence on U.S. Health Systems",
 "Trump's 2024 Plan to Lower Healthcare Costs",
 "Trump and Healthcare Funding in 2024",
 "Trump's Impact on Healthcare Innovation 2024",
 "Trump and Rural Healthcare Access 2024",
 "Trump's Healthcare Policies for Working-Class Americans",
 "Trump's Healthcare Policies and the U.S. Economy",
 "Trump's Healthcare Policy on Telemedicine 2024",
 "Trump's Impact on Healthcare Industry Jobs 2024",
 "Trump and Healthcare Insurance Premiums",
 "Trump's Healthcare Strategy for Minority Communities",
 "Trump and Healthcare Policy Changes in 2024",
 "Trump's 2024 Healthcare Plan for Seniors",
 "Trump's Efforts to Lower Prescription Drug Costs",
 "Trump's Plan for Mental Health Services in 2024",
 "Trump's Position on Healthcare for Veterans in 2024",
 "Trump's Plan for Improving Healthcare in Rural Areas",
 "Trump's Impact on Medicaid Expansion in 2024",
 "Trump's Stance on Healthcare Technology Advancements",
 "Trump's Efforts to Reduce Healthcare Fraud",
 "Trump's Strategy for Combatting Healthcare Inequality",
 "Trump's Approach to Protecting Private Health Insurance",
 "Trump and the Future of the Affordable Care Act",
 "Trump's Healthcare Policies for Families in 2024",
 "Trump's Health Insurance Plan for Small Businesses",
 "Trump's Healthcare Plan for Low-Income Families in 2024",
 "Trump's Efforts to Modernize U.S. Healthcare Systems",
 "Trump's Stance on Universal Health Coverage in 2024",
 "Trump's 2024 Healthcare Proposals for Mental Health",
 "Trump's Policy on Telemedicine Expansion",
 "Trump's Efforts to Strengthen Healthcare for Rural America"

]

Function to fetch news articles for a topic with pagination

def **fetch_news**(topic, total_articles=2000):

 all_articles = []

 page = 1

```

page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

while len(all_articles) < total_articles:
    params = {
        "q": topic,          # Query keyword
        "apiKey": API_KEY,    # API Key
        "language": "en",     # Language preference
        "sortBy": "relevance", # Sort articles by relevance
        "pageSize": page_size, # Number of articles per request
        "page": page          # Pagination page number
    }

    response = requests.get(BASE_URL, params=params)

    if response.status_code == 200:
        articles = response.json()["articles"]
        for article in articles:
            # Only extract needed fields including date
            all_articles.append({
                "title": article.get("title"),
                "description": article.get("description"),
                "url": article.get("url"),
                "publishedAt": article.get("publishedAt"),
                "source": article["source"]["name"]
            })
        print(f"Fetchd {len(articles)} articles from page {page}")
        page += 1
    else:
        print(f"Error fetching news for topic '{topic}': {response.status_code}")
        break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

    if articles:
        all_articles_data[topic] = articles
        print(f"Total articles fetched for '{topic}': {len(articles)}")
    else:
        print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("healthcare8_API_Reddit.json", "w") as json_file:

```

```
json.dump(all_articles_data, json_file, indent=4)
```

```
# Print total number of articles fetched
```

```
total_articles = sum(len(articles) for articles in all_articles_data.values())
```

```
print(f"\nTotal number of articles fetched: {total_articles}")
```

Fetching news for: Trump's 2024 Healthcare Plan: Impacts on Affordable Care Act
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's 2024 Healthcare Plan: Impacts on Affordable Care Act': 426
No articles found for 'Trump's 2024 Healthcare Plan: Impacts on Affordable Care Act'.

Fetching news for: Reforming U.S. Healthcare: Trump's Strategy for the Next Term
Fetched 0 articles from page 1
Error fetching news for topic 'Reforming U.S. Healthcare: Trump's Strategy for the Next Term': 426
No articles found for 'Reforming U.S. Healthcare: Trump's Strategy for the Next Term'.

Fetching news for: Trump's Proposed Cuts to Medicare and Medicaid: Consequences for Low-Income Americans
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Proposed Cuts to Medicare and Medicaid: Consequences for Low-Income Americans': 426
No articles found for 'Trump's Proposed Cuts to Medicare and Medicaid: Consequences for Low-Income Americans'.

Fetching news for: Trump's Impact on Healthcare Industry Regulations in 2024
Fetched 2 articles from page 1
Error fetching news for topic 'Trump's Impact on Healthcare Industry Regulations in 2024': 426
Total articles fetched for 'Trump's Impact on Healthcare Industry Regulations in 2024': 2

Fetching news for: Trump and Prescription Drug Price Reform in the U.S.
Fetched 2 articles from page 1
Error fetching news for topic 'Trump and Prescription Drug Price Reform in the U.S.': 426
Total articles fetched for 'Trump and Prescription Drug Price Reform in the U.S.': 2

Fetching news for: Healthcare Access Under Trump: Rural vs. Urban Areas
Fetched 1 articles from page 1
Error fetching news for topic 'Healthcare Access Under Trump: Rural vs. Urban Areas': 426
Total articles fetched for 'Healthcare Access Under Trump: Rural vs. Urban Areas': 1

Fetching news for: Trump's Approach to Mental Health Services and Funding
Fetched 5 articles from page 1
Error fetching news for topic 'Trump's Approach to Mental Health Services and Funding': 426
Total articles fetched for 'Trump's Approach to Mental Health Services and Funding': 5

Fetching news for: How Trump's 2024 Policies Will Affect U.S. Health Insurance Markets
Fetched 2 articles from page 1
Error fetching news for topic 'How Trump's 2024 Policies Will Affect U.S. Health Insurance Markets': 426
Total articles fetched for 'How Trump's 2024 Policies Will Affect U.S. Health Insurance Markets': 2

Fetching news for: Trump's Healthcare Vision: Balancing Private vs. Public Healthcare Systems
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Healthcare Vision: Balancing Private vs. Public Healthcare Systems': 426
No articles found for 'Trump's Healthcare Vision: Balancing Private vs. Public Healthcare Systems'.

Fetching news for: Trump Administration and Health Equity: Prospects for Minority Communities
Fetched 0 articles from page 1
Error fetching news for topic 'Trump Administration and Health Equity: Prospects for Minority Communities': 426
No articles found for 'Trump Administration and Health Equity: Prospects for Minority Communities'.

Fetching news for: The Future of U.S. Healthcare System Under Trump's Second Term
Fetched 4 articles from page 1
Error fetching news for topic 'The Future of U.S. Healthcare System Under Trump's Second Term': 426
Total articles fetched for 'The Future of U.S. Healthcare System Under Trump's Second Term': 4

Fetching news for: Trump's Stance on Reproductive Health and Women's Rights in Healthcare
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Stance on Reproductive Health and Women's Rights in Healthcare': 426
No articles found for 'Trump's Stance on Reproductive Health and Women's Rights in Healthcare'.

Fetching news for: Trump's Influence on Public Health Funding and Emergency Response Systems
Fetched 1 articles from page 1
Error fetching news for topic 'Trump's Influence on Public Health Funding and Emergency Response Systems': 426
Total articles fetched for 'Trump's Influence on Public Health Funding and Emergency Response Systems': 1

Fetching news for: Trump's Effect on the Pharmaceutical Industry: Policy Shifts and Outcomes
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Effect on the Pharmaceutical Industry: Policy Shifts and Outcomes': 426
No articles found for 'Trump's Effect on the Pharmaceutical Industry: Policy Shifts and Outcomes'.

Fetching news for: Healthcare Workforce in the Trump Administration: A Changing Landscape
Fetched 0 articles from page 1
Error fetching news for topic 'Healthcare Workforce in the Trump Administration: A Changing Landscape': 426
No articles found for 'Healthcare Workforce in the Trump Administration: A Changing Landscape'.

Fetching news for: Trump's Efforts to Combat Healthcare Fraud and Abuse in 2024
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Efforts to Combat Healthcare Fraud and Abuse in 2024': 426
No articles found for 'Trump's Efforts to Combat Healthcare Fraud and Abuse in 2024'.

Fetching news for: How Trump's Healthcare Policies Will Shape U.S. Healthcare Innovations
Fetched 0 articles from page 1
Error fetching news for topic 'How Trump's Healthcare Policies Will Shape U.S. Healthcare Innovations': 426
No articles found for 'How Trump's Healthcare Policies Will Shape U.S. Healthcare Innovations'.

Fetching news for: Trump's Impact on Healthcare Accessibility for Veterans in 2024
Fetched 0 articles from page 1
Error fetching news for topic 'Trump's Impact on Healthcare Accessibility for Veterans in 2024': 426
No articles found for 'Trump's Impact on Healthcare Accessibility for Veterans in 2024'.

Fetching news for: Will Trump's Healthcare Policies Strengthen or Weaken the Healthcare System?
Fetched 0 articles from page 1
Error fetching news for topic 'Will Trump's Healthcare Policies Strengthen or Weaken the Healthcare System?': 426

No articles found for 'Will Trump's Healthcare Policies Strengthen or Weaken the Healthcare System?'.

Fetching news for: Trump's 2024 Budget Proposals for U.S. Healthcare System: What's at Stake?

Fetches 0 articles from page 1

Error fetching news for topic 'Trump's 2024 Budget Proposals for U.S. Healthcare System: What's at Stake?': 426

No articles found for 'Trump's 2024 Budget Proposals for U.S. Healthcare System: What's at Stake?'.

Fetching news for: Trump Healthcare Policies 2024

Fetches 100 articles from page 1

Error fetching news for topic 'Trump Healthcare Policies 2024': 426

Total articles fetched for 'Trump Healthcare Policies 2024': 100

Fetching news for: Trump's Impact on U.S. Healthcare System

Fetches 3 articles from page 1

Error fetching news for topic 'Trump's Impact on U.S. Healthcare System': 426

Total articles fetched for 'Trump's Impact on U.S. Healthcare System': 3

Fetching news for: Trump's Healthcare Reform Plans 2024

Fetches 26 articles from page 1

Error fetching news for topic 'Trump's Healthcare Reform Plans 2024': 426

Total articles fetched for 'Trump's Healthcare Reform Plans 2024': 26

Fetching news for: Trump's Medicare Changes 2024

Fetches 41 articles from page 1

Error fetching news for topic 'Trump's Medicare Changes 2024': 426

Total articles fetched for 'Trump's Medicare Changes 2024': 41

Fetching news for: Trump's Health Insurance Policies

Fetches 53 articles from page 1

Error fetching news for topic 'Trump's Health Insurance Policies': 426

Total articles fetched for 'Trump's Health Insurance Policies': 53

Fetching news for: Trump and U.S. Healthcare Spending

Fetches 100 articles from page 1

Error fetching news for topic 'Trump and U.S. Healthcare Spending': 426

Total articles fetched for 'Trump and U.S. Healthcare Spending': 100

Fetching news for: Trump's Healthcare Agenda 2024

Fetches 43 articles from page 1

Error fetching news for topic 'Trump's Healthcare Agenda 2024': 426

Total articles fetched for 'Trump's Healthcare Agenda 2024': 43

Fetching news for: Trump and Health Insurance Access

Fetches 100 articles from page 1

Error fetching news for topic 'Trump and Health Insurance Access': 426

Total articles fetched for 'Trump and Health Insurance Access': 100

Fetching news for: Trump's 2024 Healthcare Budget

Fetches 37 articles from page 1
Error fetching news for topic 'Trump's 2024 Healthcare Budget': 426
Total articles fetched for 'Trump's 2024 Healthcare Budget': 37

Fetching news for: Trump's Plan for Prescription Drug Prices
Fetches 6 articles from page 1
Error fetching news for topic 'Trump's Plan for Prescription Drug Prices': 426
Total articles fetched for 'Trump's Plan for Prescription Drug Prices': 6

Fetching news for: Trump and Affordable Care Act 2024
Fetches 100 articles from page 1
Error fetching news for topic 'Trump and Affordable Care Act 2024': 426
Total articles fetched for 'Trump and Affordable Care Act 2024': 100

Fetching news for: Trump's Impact on Medicare and Medicaid
Fetches 44 articles from page 1
Error fetching news for topic 'Trump's Impact on Medicare and Medicaid': 426
Total articles fetched for 'Trump's Impact on Medicare and Medicaid': 44

Fetching news for: Trump's Healthcare Policy Changes
Fetches 40 articles from page 1
Error fetching news for topic 'Trump's Healthcare Policy Changes': 426
Total articles fetched for 'Trump's Healthcare Policy Changes': 40

Fetching news for: Trump's Views on Healthcare in 2024
Fetches 34 articles from page 1
Error fetching news for topic 'Trump's Views on Healthcare in 2024': 426
Total articles fetched for 'Trump's Views on Healthcare in 2024': 34

Fetching news for: Trump and Public Health in 2024
Fetches 100 articles from page 1
Error fetching news for topic 'Trump and Public Health in 2024': 426
Total articles fetched for 'Trump and Public Health in 2024': 100

Fetching news for: Trump's Plans for U.S. Healthcare
Fetches 76 articles from page 1
Error fetching news for topic 'Trump's Plans for U.S. Healthcare': 426
Total articles fetched for 'Trump's Plans for U.S. Healthcare': 76

Fetching news for: Trump and Healthcare for the Poor
Fetches 100 articles from page 1
Error fetching news for topic 'Trump and Healthcare for the Poor': 426
Total articles fetched for 'Trump and Healthcare for the Poor': 100

Fetching news for: Trump's Healthcare Policy for Seniors
Fetches 4 articles from page 1
Error fetching news for topic 'Trump's Healthcare Policy for Seniors': 426

Total articles fetched for 'Trump's Healthcare Policy for Seniors': 4

Fetching news for: Trump and Healthcare for Veterans

Fetched 87 articles from page 1

Error fetching news for topic 'Trump and Healthcare for Veterans': 426

Total articles fetched for 'Trump and Healthcare for Veterans': 87

Fetching news for: Trump's Healthcare Reforms

Fetched 17 articles from page 1

Error fetching news for topic 'Trump's Healthcare Reforms': 426

Total articles fetched for 'Trump's Healthcare Reforms': 17

Fetching news for: Trump's Healthcare Policy in 2024

Fetched 84 articles from page 1

Error fetching news for topic 'Trump's Healthcare Policy in 2024': 426

Total articles fetched for 'Trump's Healthcare Policy in 2024': 84

Fetching news for: Trump's Approach to Health Insurance Reform

Fetched 26 articles from page 1

Error fetching news for topic 'Trump's Approach to Health Insurance Reform': 426

Total articles fetched for 'Trump's Approach to Health Insurance Reform': 26

Fetching news for: Trump and National Healthcare Policies

Fetched 100 articles from page 1

Error fetching news for topic 'Trump and National Healthcare Policies': 426

Total articles fetched for 'Trump and National Healthcare Policies': 100

Fetching news for: Trump's Prescription Drug Pricing Plan 2024

Fetched 0 articles from page 1

Error fetching news for topic 'Trump's Prescription Drug Pricing Plan 2024': 426

No articles found for 'Trump's Prescription Drug Pricing Plan 2024'.

Fetching news for: Trump's Impact on Healthcare Access 2024

Fetched 7 articles from page 1

Error fetching news for topic 'Trump's Impact on Healthcare Access 2024': 426

Total articles fetched for 'Trump's Impact on Healthcare Access 2024': 7

Fetching news for: Trump and Health Savings Accounts (HSAs)

Fetched 1 articles from page 1

Error fetching news for topic 'Trump and Health Savings Accounts (HSAs)': 426

Total articles fetched for 'Trump and Health Savings Accounts (HSAs)': 1

Fetching news for: Trump's Vision for U.S. Healthcare 2024

Fetched 7 articles from page 1

Error fetching news for topic 'Trump's Vision for U.S. Healthcare 2024': 426

Total articles fetched for 'Trump's Vision for U.S. Healthcare 2024': 7

Fetching news for: Trump and the Affordable Care Act in 2024

Fetches 100 articles from page 1
Error fetching news for topic 'Trump and the Affordable Care Act in 2024': 426
Total articles fetched for 'Trump and the Affordable Care Act in 2024': 100

Fetching news for: Trump's Medicare for All Stance in 2024
Fetched 2 articles from page 1
Error fetching news for topic 'Trump's Medicare for All Stance in 2024': 426
Total articles fetched for 'Trump's Medicare for All Stance in 2024': 2

Fetching news for: Trump's Influence on U.S. Health Systems
Fetched 31 articles from page 1
Error fetching news for topic 'Trump's Influence on U.S. Health Systems': 426
Total articles fetched for 'Trump's Influence on U.S. Health Systems': 31

Fetching news for: Trump's 2024 Plan to Lower Healthcare Costs
Error fetching news for topic 'Trump's 2024 Plan to Lower Healthcare Costs': 429
No articles found for 'Trump's 2024 Plan to Lower Healthcare Costs'.

Fetching news for: Trump and Healthcare Funding in 2024
Error fetching news for topic 'Trump and Healthcare Funding in 2024': 429
No articles found for 'Trump and Healthcare Funding in 2024'.

Fetching news for: Trump's Impact on Healthcare Innovation 2024
Error fetching news for topic 'Trump's Impact on Healthcare Innovation 2024': 429
No articles found for 'Trump's Impact on Healthcare Innovation 2024'.

Fetching news for: Trump and Rural Healthcare Access 2024
Error fetching news for topic 'Trump and Rural Healthcare Access 2024': 429
No articles found for 'Trump and Rural Healthcare Access 2024'.

Fetching news for: Trump's Healthcare Policies for Working-Class Americans
Error fetching news for topic 'Trump's Healthcare Policies for Working-Class Americans': 429
No articles found for 'Trump's Healthcare Policies for Working-Class Americans'.

Fetching news for: Trump's Healthcare Policies and the U.S. Economy
Error fetching news for topic 'Trump's Healthcare Policies and the U.S. Economy': 429
No articles found for 'Trump's Healthcare Policies and the U.S. Economy'.

Fetching news for: Trump's Healthcare Policy on Telemedicine 2024
Error fetching news for topic 'Trump's Healthcare Policy on Telemedicine 2024': 429
No articles found for 'Trump's Healthcare Policy on Telemedicine 2024'.

Fetching news for: Trump's Impact on Healthcare Industry Jobs 2024
Error fetching news for topic 'Trump's Impact on Healthcare Industry Jobs 2024': 429
No articles found for 'Trump's Impact on Healthcare Industry Jobs 2024'.

Fetching news for: Trump and Healthcare Insurance Premiums
Error fetching news for topic 'Trump and Healthcare Insurance Premiums': 429

No articles found for 'Trump and Healthcare Insurance Premiums'.

Fetching news for: Trump's Healthcare Strategy for Minority CommunitiesTrump and Healthcare Policy Changes in 2024
Error fetching news for topic 'Trump's Healthcare Strategy for Minority CommunitiesTrump and Healthcare Policy Changes in 2024': 429

No articles found for 'Trump's Healthcare Strategy for Minority CommunitiesTrump and Healthcare Policy Changes in 2024'.

Fetching news for: Trump's 2024 Healthcare Plan for Seniors
Error fetching news for topic 'Trump's 2024 Healthcare Plan for Seniors': 429
No articles found for 'Trump's 2024 Healthcare Plan for Seniors'.

Fetching news for: Trump's Efforts to Lower Prescription Drug Costs
Error fetching news for topic 'Trump's Efforts to Lower Prescription Drug Costs': 429
No articles found for 'Trump's Efforts to Lower Prescription Drug Costs'.

Fetching news for: Trump's Plan for Mental Health Services in 2024
Error fetching news for topic 'Trump's Plan for Mental Health Services in 2024': 429
No articles found for 'Trump's Plan for Mental Health Services in 2024'.

Fetching news for: Trump's Position on Healthcare for Veterans in 2024
Error fetching news for topic 'Trump's Position on Healthcare for Veterans in 2024': 429
No articles found for 'Trump's Position on Healthcare for Veterans in 2024'.

Fetching news for: Trump's Plan for Improving Healthcare in Rural Areas

Error fetching news for topic 'Trump's Plan for Improving Healthcare in Rural Areas': 429
No articles found for 'Trump's Plan for Improving Healthcare in Rural Areas'.

Fetching news for: Trump's Impact on Medicaid Expansion in 2024
Error fetching news for topic 'Trump's Impact on Medicaid Expansion in 2024': 429
No articles found for 'Trump's Impact on Medicaid Expansion in 2024'.

Fetching news for: Trump's Stance on Healthcare Technology Advancements
Error fetching news for topic 'Trump's Stance on Healthcare Technology Advancements': 429
No articles found for 'Trump's Stance on Healthcare Technology Advancements'.

Fetching news for: Trump's Efforts to Reduce Healthcare Fraud
Error fetching news for topic 'Trump's Efforts to Reduce Healthcare Fraud': 429
No articles found for 'Trump's Efforts to Reduce Healthcare Fraud'.

Fetching news for: Trump's Strategy for Combatting Healthcare Inequality
Error fetching news for topic 'Trump's Strategy for Combatting Healthcare Inequality': 429
No articles found for 'Trump's Strategy for Combatting Healthcare Inequality'.

Fetching news for: Trump's Approach to Protecting Private Health Insurance
Error fetching news for topic 'Trump's Approach to Protecting Private Health Insurance': 429
No articles found for 'Trump's Approach to Protecting Private Health Insurance'.

Fetching news for: Trump and the Future of the Affordable Care Act
Error fetching news for topic 'Trump and the Future of the Affordable Care Act': 429
No articles found for 'Trump and the Future of the Affordable Care Act'.

Fetching news for: Trump's Healthcare Policies for Families in 2024
Error fetching news for topic 'Trump's Healthcare Policies for Families in 2024': 429
No articles found for 'Trump's Healthcare Policies for Families in 2024'.

Fetching news for: Trump's Health Insurance Plan for Small Businesses
Error fetching news for topic 'Trump's Health Insurance Plan for Small Businesses': 429
No articles found for 'Trump's Health Insurance Plan for Small Businesses'.

Fetching news for: Trump's Healthcare Plan for Low-Income Families in 2024
Error fetching news for topic 'Trump's Healthcare Plan for Low-Income Families in 2024': 429
No articles found for 'Trump's Healthcare Plan for Low-Income Families in 2024'.

Fetching news for: Trump's Efforts to Modernize U.S. Healthcare Systems
Error fetching news for topic 'Trump's Efforts to Modernize U.S. Healthcare Systems': 429
No articles found for 'Trump's Efforts to Modernize U.S. Healthcare Systems'.

Fetching news for: Trump's Stance on Universal Health Coverage in 2024
Error fetching news for topic 'Trump's Stance on Universal Health Coverage in 2024': 429
No articles found for 'Trump's Stance on Universal Health Coverage in 2024'.

Fetching news for: Trump's 2024 Healthcare Proposals for Mental Health
Error fetching news for topic 'Trump's 2024 Healthcare Proposals for Mental Health': 429
No articles found for 'Trump's 2024 Healthcare Proposals for Mental Health'.

Fetching news for: Trump's Policy on Telemedicine Expansion
Error fetching news for topic 'Trump's Policy on Telemedicine Expansion': 429
No articles found for 'Trump's Policy on Telemedicine Expansion'.

Fetching news for: Trump's Efforts to Strengthen Healthcare for Rural America
Error fetching news for topic 'Trump's Efforts to Strengthen Healthcare for Rural America': 429
No articles found for 'Trump's Efforts to Strengthen Healthcare for Rural America'.

Total number of articles fetched: 1486

```

In [20]: import json
from datetime import datetime

# File paths
input_file = 'healthcare8_API_Reddit.json'
output_file = 'cleaned_healthcare8.json'

# Load the JSON data
with open(input_file, 'r', encoding='utf-8') as file:
    data = json.load(file)

# Process the JSON data
for category, articles in data.items():
    for article in articles:
        # Rename 'publishedAt' to 'date' and clean it
        published_at = article.pop("publishedAt", None)
        if published_at:
            try:
                # Convert and format date
                clean_date = datetime.fromisoformat(published_at.replace("Z", "+00:00")).strftime("%Y-%m-%d %H:%M:%S")
                article["date"] = clean_date
            except ValueError:
                print(f"Error parsing date: {published_at}")
                article["date"] = published_at # Keep the original if parsing fails

# Save the cleaned data
with open(output_file, 'w', encoding='utf-8') as file:
    json.dump(data, file, indent=4)

print(f"Cleaned data saved to {output_file}")

```

Cleaned data saved to cleaned_healthcare8.json

In [21]: **import** json

```
# Load the JSON file
file_path = "cleaned_healthcare8.json" # Update with the correct file path
with open(file_path, 'r') as file:
    data = json.load(file)

# Assuming the articles or topics are in a dictionary, convert them into a list
articles_list = []

# Example of extracting data into a list, adjust key names as per the JSON structure
if isinstance(data, dict):
    for key, value in data.items():
        if isinstance(value, str): # If the value is a string (e.g., an article or topic)
            articles_list.append(value)
        elif isinstance(value, list): # If the value itself is a list of articles
            articles_list.extend(value)

# If the data is already a list of strings, you can simply assign it
if isinstance(data, list) and all(isinstance(item, str) for item in data):
    articles_list = data

# Print the list or save it to a new JSON file
print(articles_list)

# Save back as a list in a new JSON file
output_path = "cleaned_healthcare8_list.json"
with open(output_path, 'w') as output_file:
    json.dump(articles_list, output_file, indent=4)
print(f"Saved the list to {output_path}")
```

[{'title': 'Why Trump's Return Won't Halt Progress On Clean Energy, Says Industry', 'description': 'While Trump's return may spur attempts to undo Biden-era clean energy initiatives, the transition is set to continue—driven by innovation, regulations, and market forces.', 'url': 'https://www.forbes.com/sites/globalcitizen/2024/12/16/why-trumps-return-wont-halt-progress-on-clean-energy-says-industry/', 'source': 'Forbes', 'date': '2024-12-16 13:30:38'}, {'title': '2:00PM Water Cooler 12/11/2024', 'description': '~ Today's Water Cooler: Politics, syndemics ~', 'url': 'https://www.nakedcapitalism.com/2024/12/200pm-water-cooler-12-11-2024.html', 'source': 'Nakedcapitalism.com', 'date': '2024-12-11 19:00:20'}, {'title': 'The Progressive Case for RFK Jr. | Opinion', 'description': 'Americans should support Kennedy's nomination. He is the best chance of reigning in corruption and corporate power while prioritizing public health over profit.', 'url': 'https://www.newsweek.com/progressive-case-rfk-jr-opinion-1992709', 'source': 'Newsweek', 'date': '2024-11-27 19:51:53'}, {'title': 'Can Trump Make America Healthy Again?', 'description': 'By Neenah Payne Wendell Berry: “We have a food system that ignores health and a healthcare system that ignores food!” Hippocrates (“Founder” of Western Medicine):...\\nCan Trump Make America Healthy Again?', 'url': 'https://activistpost.com/2024/12/can-trump-make-america-healthy-again.html', 'source': 'Activistpost.com', 'date': '2024-12-04 12:10:42'}, {'title': 'Keeping Score: More Attacks on Trans Rights; Sexual Assault Should Disqualify Cabinet Nominees, Americans Say; Female Professors Win Lawsuit and Backpay for Pay Disparity', 'description': 'In every issue of Ms., we track research on our progress in the fight for equality, catalogue can't-miss quotes from feminist voices and keep tabs on the feminist movement's many milestones. We're Keeping Score online, too—in this biweekly roundup.\\nThis week:...', 'url': 'https://msmagazine.com/2024/12/11/feminist-news-keeping-score-trans-skrmetti-equal-pay-trump-cabinet/', 'source': 'Ms. Magazine', 'date': '2024-12-11 18:53:20'}, {'title': 'Hegseth fights to save Pentagon nomination as sources say


```
In [19]: import requests
import json

# Your API key
API_KEY = "250c39188e144bc09d574051ce483b94"
BASE_URL = "https://newsapi.org/v2/everything"

# Topics to search for related to Trump's presidency and defense
topics = [
    "Trump's Policies on Healthcare for Undocumented Immigrants",
    "Trump's 2024 Plans to Address the Opioid Crisis in the U.S.",
    "Impact of Trump's Healthcare Policies on LGBTQ+ Communities",
    "Trump's Vision for Healthcare Digitalization and AI Integration",
    "Trump's Influence on State-Level Medicaid Programs",
    "Trump's Approach to Tackling Health Disparities Across the U.S.",
    "Trump's Plans for Long-Term Care and Aging Population",
    "Trump's 2024 Vision for Integrating Mental and Physical Healthcare",
    "Impact of Trump's Policies on Employer-Sponsored Health Insurance",
    "Trump's Role in Reshaping Preventative Healthcare Initiatives",
    "How Trump's 2024 Healthcare Policies Will Affect Medical Research Funding",
    "Trump's Stance on Healthcare Trade Deals Post-2024",
    "Impact of Trump's Healthcare Policies on Drug Manufacturing in the U.S.",
    "Trump's 2024 Healthcare Vision for Children and Adolescents",
    "Trump's Policies on Vaccination and Public Health Campaigns",
    "Trump's Strategy to Reduce Hospital Costs and Administrative Burdens",
    "The Role of Trump's Policies in Shaping Telehealth Accessibility",
    "Trump's Plans to Address the Nursing Shortage in the U.S.",
    "Impact of Trump's 2024 Healthcare Proposals on Chronic Disease Management",
    "Trump's Efforts to Reform Healthcare Pricing Transparency",
    "Trump's Policies on Cross-Border Healthcare Services",
    "Trump's Focus on Health Data Privacy and Cybersecurity in Healthcare",
    "How Trump's Healthcare Policies Influence Pharmaceutical Pricing Abroad",
    "Trump's Vision for Healthcare Collaboration Between Federal and State Governments",
    "Trump's Approach to Reducing Maternal Mortality Rates in the U.S.",
    "Trump's Plan for Reducing Healthcare Debt for Americans",
    "Trump's Policies on Alternative Medicine and Holistic Healthcare",
    "How Trump's Healthcare Reforms Affect Medical Device Innovation",
    "Trump's Stance on Expanding Healthcare Access in Indigenous Communities",
    "Trump's Policies on Reducing Red Tape in Healthcare Regulations",
    "Trump's Role in Reshaping U.S. Healthcare Provider Networks",
    "Trump's Policies on Addressing Healthcare Burnout Among Professionals",
    "Trump's Plans to Expand Military Healthcare Services for Active Personnel",
    "Trump's Impact on Healthcare Technology Startups in the U.S.",
    "Trump's Role in Negotiating International Drug Price Agreements",
    "Trump's Vision for Universal Drug Coverage in the U.S.",
    "Trump's 2024 Healthcare Policies and Their Effect on Urban Hospitals",
    "Impact of Trump's Proposals on Specialty Healthcare Providers",
```

```

    "Trump's Focus on Healthcare Policy for Disaster Relief and Emergencies",
    "How Trump's Healthcare Reforms Will Influence Medical Education in the U.S."
]

# Function to fetch news articles for a topic with pagination
def fetch_news(topic, total_articles=2000):
    all_articles = []
    page = 1
    page_size = 100 # Maximum number of articles per request (NewsAPI Limit)

    while len(all_articles) < total_articles:
        params = {
            "q": topic, # Query keyword
            "apiKey": API_KEY, # API Key
            "language": "en", # Language preference
            "sortBy": "relevance", # Sort articles by relevance
            "pageSize": page_size, # Number of articles per request
            "page": page # Pagination page number
        }

        response = requests.get(BASE_URL, params=params)

        if response.status_code == 200:
            articles = response.json()["articles"]
            for article in articles:
                # Only extract needed fields including date
                all_articles.append({
                    "title": article.get("title"),
                    "description": article.get("description"),
                    "url": article.get("url"),
                    "publishedAt": article.get("publishedAt"),
                    "source": article["source"]["name"]
                })
            print(f"Fetches {len(articles)} articles from page {page}")
            page += 1
        else:
            print(f"Error fetching news for topic '{topic}': {response.status_code}")
            break

    return all_articles[:total_articles] # Return up to the desired number of articles

# Fetch news for each topic and save the output in a JSON file
all_articles_data = {}

for topic in topics:
    print(f"\nFetching news for: {topic}")
    articles = fetch_news(topic)

```

```

if articles:
    all_articles_data[topic] = articles
    print(f"Total articles fetched for '{topic}': {len(articles)}")
else:
    print(f"No articles found for '{topic}'.")

# Save the collected articles to a JSON file
with open("healthcare9_API_Reddit.json", "w") as json_file:
    json.dump(all_articles_data, json_file, indent=4)

# Print total number of articles fetched
total_articles = sum(len(articles) for articles in all_articles_data.values())
print(f"\nTotal number of articles fetched: {total_articles}")

```

Fetching news for: Trump's Policies on Healthcare for Undocumented Immigrants
 Error fetching news for topic 'Trump's Policies on Healthcare for Undocumented Immigrants': 429
 No articles found for 'Trump's Policies on Healthcare for Undocumented Immigrants'.

Fetching news for: Trump's 2024 Plans to Address the Opioid Crisis in the U.S.
 Error fetching news for topic 'Trump's 2024 Plans to Address the Opioid Crisis in the U.S.': 429
 No articles found for 'Trump's 2024 Plans to Address the Opioid Crisis in the U.S.'.

Fetching news for: Impact of Trump's Healthcare Policies on LGBTQ+ Communities
 Error fetching news for topic 'Impact of Trump's Healthcare Policies on LGBTQ+ Communities': 429
 No articles found for 'Impact of Trump's Healthcare Policies on LGBTQ+ Communities'.

Fetching news for: Trump's Vision for Healthcare Digitalization and AI Integration
 Error fetching news for topic 'Trump's Vision for Healthcare Digitalization and AI Integration': 429
 No articles found for 'Trump's Vision for Healthcare Digitalization and AI Integration'.

Fetching news for: Trump's Influence on State-Level Medicaid Programs
 Error fetching news for topic 'Trump's Influence on State-Level Medicaid Programs': 429
 No articles found for 'Trump's Influence on State-Level Medicaid Programs'.


```

In [16]: import requests
import json
from datetime import datetime, timedelta
import re
import calendar

# SerpAPI key
api_key = "d050e401f6383d477f0f145af4c5d1339acf34ffbf70518d4dba737661887a22"

# List of topics
topics = [
    "Trump's Healthcare Reform Plans 2024",
    "Trump's Impact on Affordable Care Act (ACA)",
    "Trump's Approach to Health Insurance Premiums in 2024",
    "Trump Administration and Medicare for All",
    "Trump's Proposals for Prescription Drug Pricing in 2024",
    "Trump's Strategy to Lower Healthcare Costs in America",
    "Trump and Medicaid Expansion in 2024",
    "Trump's Stance on Healthcare for Veterans",
    "Trump's Policy on Healthcare Access in Rural Areas",
    "Trump's Vision for U.S. Health Insurance Market in 2024",
    "Trump's Policies on Healthcare for Senior Citizens",
    "Trump's Influence on Mental Health Care in the U.S.",
    "Trump's Healthcare Policies and Minority Populations",
    "Trump's Vision for Healthcare in America: 2024 and Beyond",
    "Trump's Policies on Healthcare Technology and Innovation",
    "Trump's Impact on Health Insurance for the Uninsured",
    "Trump and Public Health Crises: Preparing for the Future",
    "Trump's Approach to Expanding Telemedicine in the U.S.",
    "Trump's Stance on Vaccination and Public Health in 2024",
    "Trump's Impact on Healthcare Providers and Hospitals in 2024"
]

# Define the output file name
output_file = "healthcare9_API.json"

# Function to fetch news articles from SerpAPI
def fetch_news(topic):
    url = f"https://serpapi.com/search"
    params = {
        "q": topic,
        "api_key": api_key,
        "tbm": "nws", # News results
    }
    response = requests.get(url, params=params)
    return response.json()

```

```

# Function to convert relative date to exact date
def convert_to_exact_date(relative_date):
    # Match time formats like "1 day ago", "3 hours ago", "1 month ago", etc.
    today = datetime.today()

    # Days ago
    days_ago_match = re.match(r'(\d+)\s*day[s]*\s*ago', relative_date)
    # Hours ago
    hours_ago_match = re.match(r'(\d+)\s*hour[s]*\s*ago', relative_date)
    # Minutes ago
    minutes_ago_match = re.match(r'(\d+)\s*minute[s]*\s*ago', relative_date)
    # Weeks ago
    weeks_ago_match = re.match(r'(\d+)\s*week[s]*\s*ago', relative_date)
    # Months ago
    months_ago_match = re.match(r'(\d+)\s*month[s]*\s*ago', relative_date)
    # Years ago
    years_ago_match = re.match(r'(\d+)\s*year[s]*\s*ago', relative_date)

    if days_ago_match:
        days_ago = int(days_ago_match.group(1))
        return today - timedelta(days=days_ago)
    elif hours_ago_match:
        hours_ago = int(hours_ago_match.group(1))
        return today - timedelta(hours=hours_ago)
    elif minutes_ago_match:
        minutes_ago = int(minutes_ago_match.group(1))
        return today - timedelta(minutes=minutes_ago)
    elif weeks_ago_match:
        weeks_ago = int(weeks_ago_match.group(1))
        return today - timedelta(weeks=weeks_ago)
    elif months_ago_match:
        months_ago = int(months_ago_match.group(1))
        new_month = today.month - months_ago
        if new_month <= 0:
            new_month += 12
            new_year = today.year - 1
        else:
            new_year = today.year
        # Handle month overflows, i.e., going back to previous months
        return today.replace(year=new_year, month=new_month)
    elif years_ago_match:
        years_ago = int(years_ago_match.group(1))
        return today.replace(year=today.year - years_ago)
    else:
        # If no match, return the date as is (might be in a format like '2024-12-12')
        return relative_date

```

```

# Initialize an empty list to store the articles

```

```

all_articles = []

# Iterate over each topic, fetch news articles, and add them to the list
for topic in topics:
    print(f"Fetching articles for: {topic}")
    data = fetch_news(topic)

    # Check if 'news_results' exists in the response
    if "news_results" in data:
        for article in data["news_results"]:
            # Convert the relative date to the exact date
            published_at = article.get("date")
            exact_date = convert_to_exact_date(published_at)
            if isinstance(exact_date, datetime):
                exact_date = exact_date.strftime("%Y-%m-%d") # Format as 'YYYY-MM-DD'

            article_data = {
                "title": article.get("title"),
                "description": article.get("snippet"),
                "url": article.get("link"),
                "publishedAt": exact_date, # Use the exact date
                "source": article.get("source"),
                "image": article.get("source_info", {}).get("image"),
            }
            all_articles.append(article_data)

# Save the fetched data to a JSON file
with open(output_file, "w") as file:
    json.dump(all_articles, file, indent=4)

print(f"Fetched articles have been saved to {output_file}")

```

Fetching articles for: Trump's Healthcare Reform Plans 2024
Fetching articles for: Trump's Impact on Affordable Care Act (ACA)
Fetching articles for: Trump's Approach to Health Insurance Premiums in 2024
Fetching articles for: Trump Administration and Medicare for All
Fetching articles for: Trump's Proposals for Prescription Drug Pricing in 2024
Fetching articles for: Trump's Strategy to Lower Healthcare Costs in America
Fetching articles for: Trump and Medicaid Expansion in 2024
Fetching articles for: Trump's Stance on Healthcare for Veterans
Fetching articles for: Trump's Policy on Healthcare Access in Rural Areas
Fetching articles for: Trump's Vision for U.S. Health Insurance Market in 2024
Fetching articles for: Trump's Policies on Healthcare for Senior Citizens
Fetching articles for: Trump's Influence on Mental Health Care in the U.S.
Fetching articles for: Trump's Healthcare Policies and Minority Populations
Fetching articles for: Trump's Vision for Healthcare in America: 2024 and Beyond
Fetching articles for: Trump's Policies on Healthcare Technology and Innovation
Fetching articles for: Trump's Impact on Health Insurance for the Uninsured
Fetching articles for: Trump and Public Health Crises: Preparing for the Future
Fetching articles for: Trump's Approach to Expanding Telemedicine in the U.S.
Fetching articles for: Trump's Stance on Vaccination and Public Health in 2024
Fetching articles for: Trump's Impact on Healthcare Providers and Hospitals in 2024
Fetched articles have been saved to healthcare9_API.json

In [17]: `import json`

```
# List of files to update
input_files = ["healthcare8_API.json", "healthcare9_API.json"]

# Iterate over each file and process
for input_file in input_files:
    with open(input_file, "r") as file:
        # Load the existing data from the file
        data = json.load(file)

        # Update the "publishedAt" field to "date"
        for article in data:
            if "publishedAt" in article:
                article["date"] = article.pop("publishedAt") # Change the key name to "date"

        # Save the updated data back to the same file
        with open(input_file, "w") as file:
            json.dump(data, file, indent=4)

    print(f"Updated {input_file} with 'date' key instead of 'publishedAt'.")
```

Updated healthcare8_API.json with 'date' key instead of 'publishedAt'.

Updated healthcare9_API.json with 'date' key instead of 'publishedAt'.