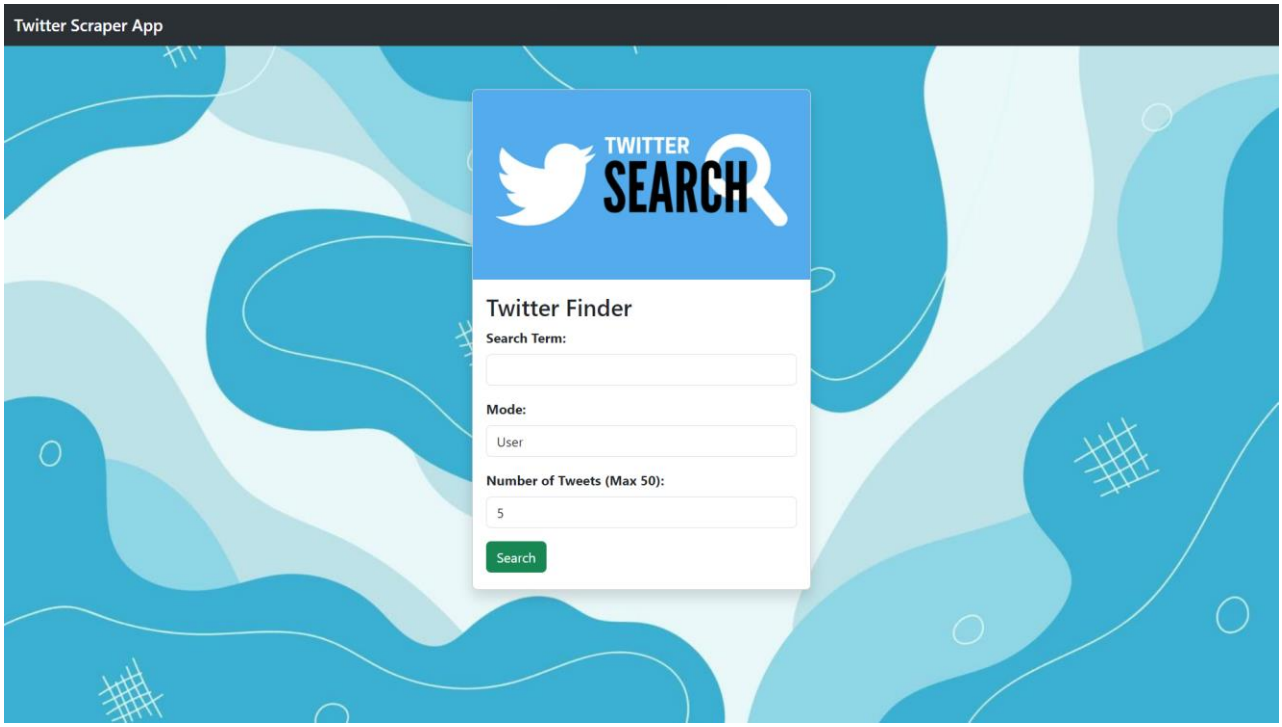
## Open Source Programming Club

Hrishikesh Kulkarni - 21BAI1660

Tejas Mandloi – 21BAI1227

**Problem Statement -** Extracting comments from Twitter on a specified topic using Python**.**

# Twitter Scraper App




The Twitter scraper app demonstrates an implementation of a web scraping application for extracting comments from Twitter on a specified topic. The script's primary components involve utilizing the **Flask web framework** and a custom web scraping library, named **ntscraper**, which interacts with the Nitter platform—an alternative front-end for Twitter.

Here's a brief description of the script's functionality using Flask:

1. **Flask Setup:**
    - The script initializes a Flask web application (**Flask(__name__)**) to create a simple web interface.
2. **Web Scraping Function:**
    - The **get_tweets** function is defined to extract tweets and user data from Twitter using the **ntscraper** library.
    - The function takes parameters such as the Twitter username (**name**), the mode of operation (**mode**), and the number of tweets to retrieve (**no**).
3. **Data Processing:**
    - Inside the **get_tweets** function, the **Nitter** scraper is initialized, and tweets are retrieved based on the specified parameters.
    - The retrieved tweet data is processed to extract relevant information, such as tweet text, date, likes, comments, and a link to the tweet.
    - User data is optionally retrieved if the mode is set to 'user,' including information like username, name, bio, image, followers, and following count.
4. **Error Handling:**
    - The script includes error handling within the **get_tweets** function to capture and print any exceptions that might occur during the scraping process. An error message is generated and returned if an exception occurs.
5. **Flask Routes:**
    - Two Flask routes are defined:
        - The **'/'** route renders the main page (assumed to be 'index.html').
        - The **'/search'** route is set up to handle form submissions via POST method. It extracts form data (Twitter username, mode, and number of tweets) and calls the **get_tweets** function. The results are then passed to a template ('result.html') for rendering.
6. **Script Execution:**
    - The script includes a block (**if __name__ == '__main__':**) to start the Flask application in debug mode if the script is executed directly.

# Code –

## Web Scrapping using Flask

## app.py

```
# Importing necessary libraries and modules

from flask import Flask, render_template, request, redirect, url_for
import pandas as pd
from ntscraper import Nitter  # Importing necessary modules

app = Flask(__name__)

def get_tweets(name, mode, no):
    try:
        # Initialize Nitter scraper
        scraper = Nitter()

        # Get tweets based on the input parameters
        tweets = scraper.get_tweets(name, mode=mode, number=no)
        final_tweets = []

        # Extract relevant information from the tweets
        for x in tweets['tweets']:
            date_without_time = pd.to_datetime(x['date'], format='%b %d, %Y · 
%I:%M %p UTC').strftime('%b %d, %Y')
            data = [x['link'], x['text'], date_without_time, 
x['stats']['likes'], x['stats']['comments']]
            final_tweets.append(data)

        # Create a DataFrame from the extracted tweet information
        df = pd.DataFrame(final_tweets, columns=['twitter_link', 'text', 
'date', 'likes', 'comments'])

        # Additional code for user_data retrieval
        user_data = None
        if mode == 'user':
            profile_info = scraper.get_profile_info(name)
            user_data = {
                'username': profile_info['username'],
                'name': profile_info['name'],
                'bio': profile_info['bio'],
                'image': profile_info['image'],
                'followers': profile_info['stats']['followers'],
                'following': profile_info['stats']['following']
            }
      # user_data object contains all the information about the entered
```

```
        term

            return df, user_data, None  # No error message

    except Exception as e:
        # Handle exceptions and provide an error message
        error_message = f"An Unknown error occurred: {e}. Please try again. Or
try another, Eg. User (UserID to be entered in the search term) - imVkohli,
BillGates etc"
        print(error_message)
        return pd.DataFrame(), None, error_message

# Creating Routes for the Application

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/search', methods=['POST'])
def search():
    # Extract search parameters from the form
    name = request.form['name']
    mode = request.form['mode']
    no = int(request.form['no'])

    # Get tweets and user data using the search parameters
    result_tuple = get_tweets(name, mode, no)
    tweets_df, user_data, error_message = result_tuple

    # Render the result template with the obtained data
    return render_template('result.html', name=name, mode=mode, no=no,
tweets_df=tweets_df, user_data=user_data, error_message=error_message)

if __name__ == '__main__':
    # Run the Flask application in debug mode
    app.run(debug=True)
```

## Front-end using HTML, CSS and JavaScript

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Metadata for character set and viewport -->
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>App</title>

    <!-- Bootstrap CSS and custom stylesheet -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
        rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
        crossorigin="anonymous">
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">
</head>

<body class="d-flex flex-column min-vh-100">
    <!-- Navbar section -->
    <nav class="navbar navbar-expand-lg bg-body-tertiary sticky-top" data-bs-
theme="dark">
        <div class="container-fluid">
          <span class="navbar-brand mb-0 h1">Twitter Scraper App</span>
        </div>
    </nav>

    <!-- Main content section -->
    <main class="mainclass container mt-1">
        <div class="container d-flex justify-content-center align-items-center
mt-5">
            <div class="row flex-fill">
                <div class="col-md-6 offset-md-3 col-xl-4 offset-xl-4">
                    <!-- Card containing search form -->
                    <div class="card shadow">
                        <img
src="https://i0.wp.com/softwareengineeringdaily.com/wp-
content/uploads/2020/09/TWITTER-SEARCH-1280x720-1.jpg?fit=1280%2C720" alt=""
class="card-img-top">
                        <div class="card-body">
                            <h3 class="card-title">Twitter Finder</h3>
                            <!-- Search form -->
                            <form action="/search" method="post"
onsubmit="showLoading()">
                                <div class="mb-3">
                                    <label for="name" class="form-
label"><strong>Search Term:</strong></label>
                                    <input type="text" class="form-control"
id="name" name="name" required>
                                </div>
                                <div class="mb-3">
                                    <label for="mode" class="form-
label"><strong>Mode:</strong></label>
```

```
                                    <select id="mode" name="mode" class="form-
control" required>
                                        <option value="user">User</option>
                                        <option
value="hashtag">Hashtag</option>
                                        <option value="term">Term</option>
                                    </select>
                                </div>
                                <div class="mb-3">
                                    <label for="no" class="form-
label"><strong>Number of Tweets (Max 50):</strong></label>
                                    <input type="number" id="no" name="no"
min="1" max="50" value="5" class="form-control">
                                </div>
                                <div class="mb-1">
                                    <button class="btn btn-success"
type="submit">Search</button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </main>

    <!-- Loading indicator section -->
    <div id="loading" class="text-center" style="display: none;">
        <img id='loading-gif'
src="https://media2.giphy.com/media/RgzryV9nRCMHPVVXPV/giphy.gif?cid=ecf05e47f
27ps7ic8s7mevfgrdpwsnqdrpsc6348mr6l40az&ep=v1_gifs_search&rid=giphy.gif&ct=g"
alt="">
        <p>Loading... This may take a moment, please be patient<br>It should
be done in 2-5 minutes.</p>
    </div>

    <!-- JavaScript functions for showing loading indicator -->
    <script>
        function showLoading() {
            document.querySelector('.mainclass').style.display = 'none';
            document.getElementById("loading").style.display = "block";
            document.body.classList.add('black-bg');
        }
    </script>

    <!-- jQuery and Bootstrap JavaScript -->
```

```
    <script src="https://code.jquery.com/jquery-3.7.1.min.js"
integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs0OEvtFFmqPF/lYI/Cxo="
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min
.js"
        integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
        crossorigin="anonymous"></script>
</body>
</html>
```

This HTML document employs the Bootstrap framework to structure and style the web page for the "Twitter Scraper App." The page features a responsive navigation bar, a central card component for the search form, and a loading section with a GIF for enhanced user experience. The form includes fields for the search term, mode selection (user, hashtag, term), and the number of tweets to retrieve. Upon submission, a JavaScript function toggles the display of the main content and the loading section. External scripts for jQuery and Bootstrap enhance dynamic functionality. Overall, the design prioritizes user-friendly interaction, combining a visually appealing layout with practical features for efficient Twitter data retrieval.

Result.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Meta tags for character set and viewport -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
        rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
        crossorigin="anonymous">

    <!-- Custom CSS -->
```

```
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">

    <!-- Page title -->
    <title>Search Result</title>
</head>

<body class="d-flex flex-column min-vh-100">
    <!-- Navigation bar -->
    <nav class="navbar navbar-expand-lg bg-body-tertiary sticky-top" data-bs-
theme="dark">
        <div class="container-fluid">
            <span class="navbar-brand mb-0 h1">Twitter Scraper App</span>
        </div>
    </nav>

    <main>
        <!-- Main content container -->
        <div class="container mt-4">
            <!-- Back to Search Page button -->
            <p><a href="{{ url_for('index') }}"><button class="btn btn-
secondary">Back to Search Page</button></a></p>

            <!-- Display error message if any -->
            {% if error_message %}
                <p>{{ error_message }}</p>
            {% endif %}

            <!-- Display user details if available -->
            {% if user_data %}
                <div class="container mt-4 mb-2">
                    <div class="row">
                        <!-- Column for user details -->
                        <div class="col-md-6">
                            <h3 class="mb-4">User Details</h3>
                            <!-- User details information -->
                        </div>
                        <!-- Column for user image -->
                        <div class="col-md-6 text-center">
                            <!-- User image -->
                        </div>
                    </div>
                </div>
            {% endif %}
            <!-- Display search result information -->
            <h2 class="mb-3">Search Result for {{ name }}</h2>
            <h6>Mode: {{ mode }}</h6>
            <h6>Number of Tweets: {{ no }}</h6>
```

```
            <!-- Display the DataFrame in a Bootstrap-styled table -->
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th>Twitter Link</th>
                        <th>Text</th>
                        <th>Date</th>
                        <th>Likes</th>
                        <th>Comments</th>
                    </tr>
                </thead>
                <tbody>
                    <!-- Loop through each tweet and display information -->
                    {% for index, row in tweets_df.iterrows() %}
                        <tr>
                            <td><a href="{{ row['twitter_link'] }}"
target="_blank">{{ row['twitter_link'] }}</a></td>
                            <td>{{ row['text'] }}</td>
                            <td>{{ row['date'] }}</td>
                            <td>{{ row['likes'] }}</td>
                            <td>{{ row['comments'] }}</td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </main>

    <!-- JavaScript for showing loading state -->
    <script>
        function showLoading() {

            // Show loading elements and hide content
            document.getElementById("loading").style.display = "block";
            document.getElementById("content").style.display = "none";
            document.getElementById("loadingGif").style.display = "block";
        }

       // Add event listener to the search form to trigger loading state
        document.getElementById("searchForm").addEventListener("submit",
function () {
            showLoading();
        });

    </script>
```

```
    <!-- jQuery and Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.7.1.min.js"
        integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs0OEvtFFmqPF/lYI/Cxo="
crossorigin="anonymous"></script>

      <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min
.js"
        integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
        crossorigin="anonymous"></script>


</body>

</html>
```

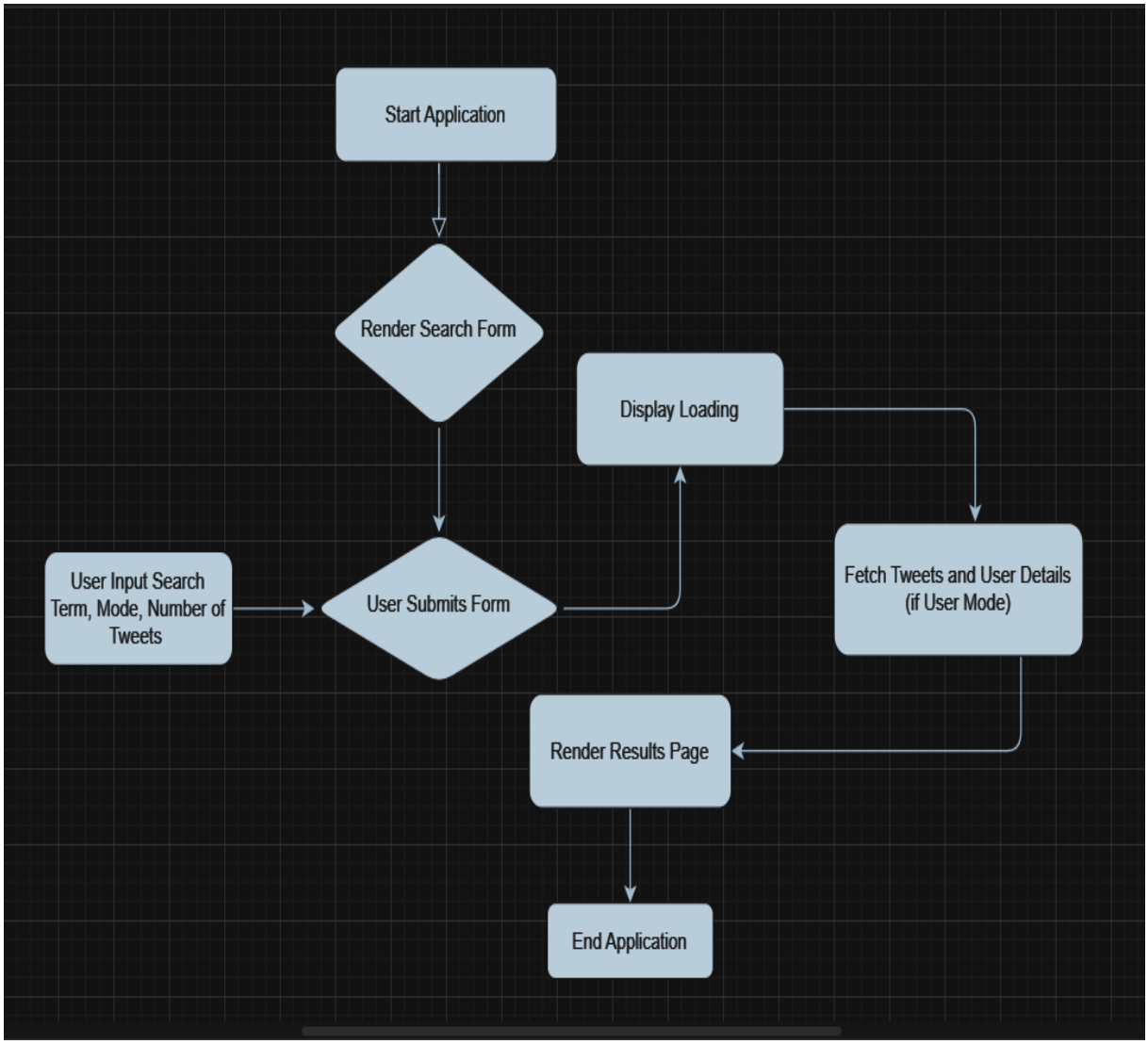
In this HTML document, Bootstrap, a popular front-end framework, has been utilized to enhance the visual presentation and responsiveness of the Twitter Scraper App's search result page. The use of Bootstrap's predefined classes ensures a consistent and well-designed layout.

To improve user experience during the search process, a loading page is implemented. JavaScript is employed to toggle the visibility of elements—displaying a loading indicator while hiding the main content. This creates a smooth transition between the loading state and the eventual display of search results.

Additionally, the search results, originally in Python DataFrame format, are seamlessly transformed into a visually appealing table using Bootstrap's table classes. This provides a structured and organized presentation of tweet information, including details such as Twitter link, text, date, likes, and comments. The responsive design ensures that the table adjusts gracefully to different screen sizes.

Notably, the card features a dynamic profile picture fetched from Flask. Bootstrap classes are utilized for card styling, ensuring an aesthetically pleasing design. The addition of a profile image enhances visual appeal and user engagement, contributing to an overall well-designed interface for efficient Twitter data retrieval.

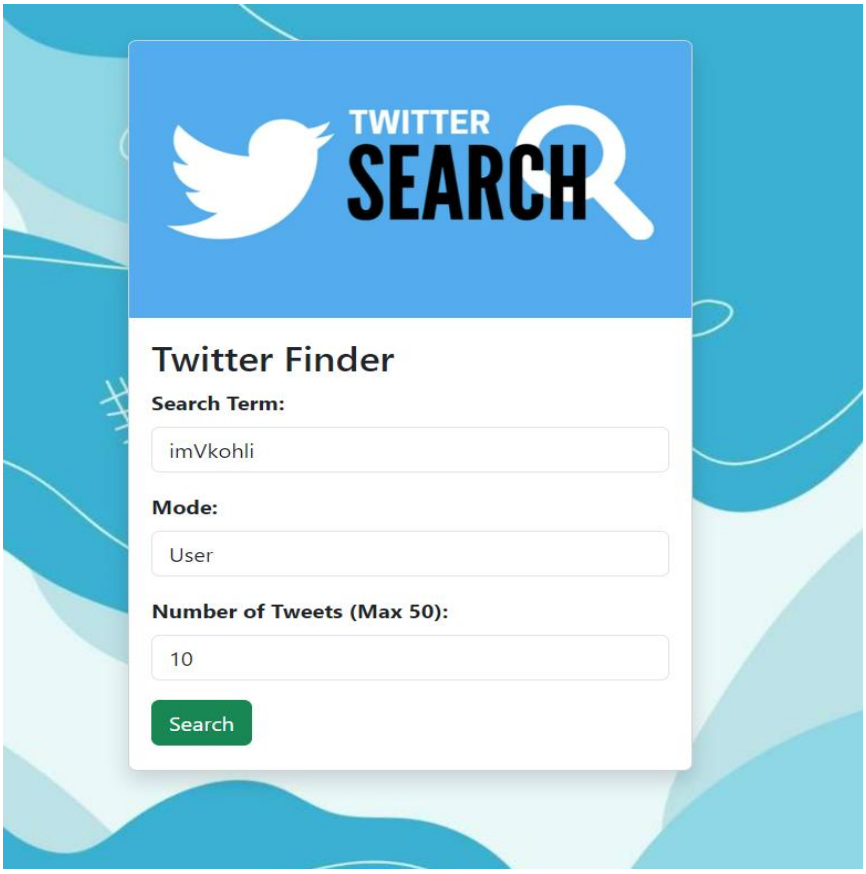
## Flowchart –



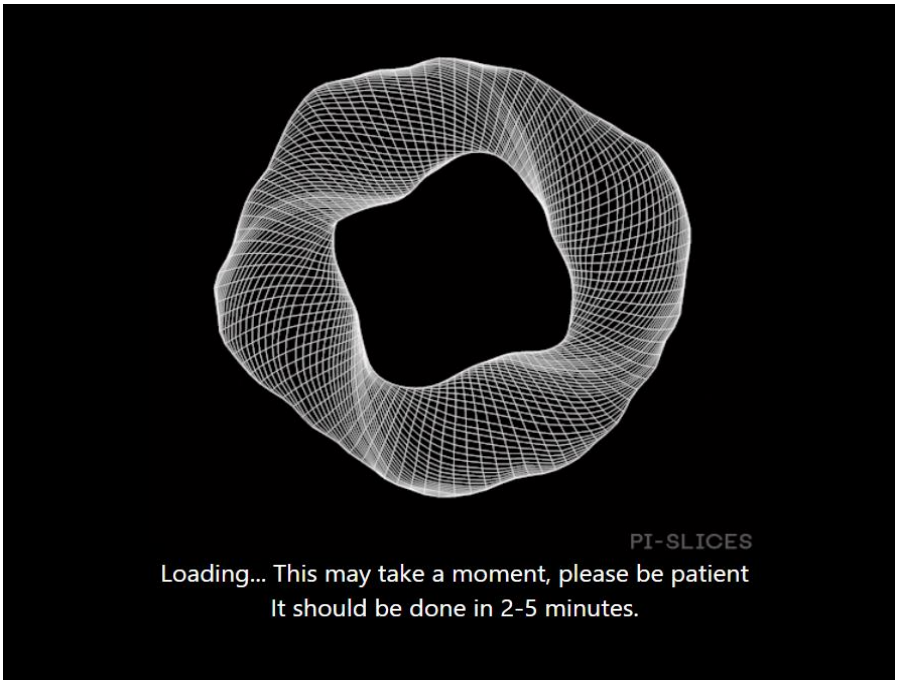

The application begins by rendering a search form, presenting users with fields to input their search terms, choosing a mode ('user' or 'search'), and specifying the number of tweets they want. Once the user submits the form, the server processes the inputs, triggering the retrieval of tweets and user details using web scraping. During this process, a loading screen is displayed to signal ongoing data retrieval. After completion, the server renders a results page showcasing the fetched tweets and, if applicable, additional user details. The application concludes its execution, providing users with the desired information based on their input parameters.

**Output 1 ( Mode : User ) –**

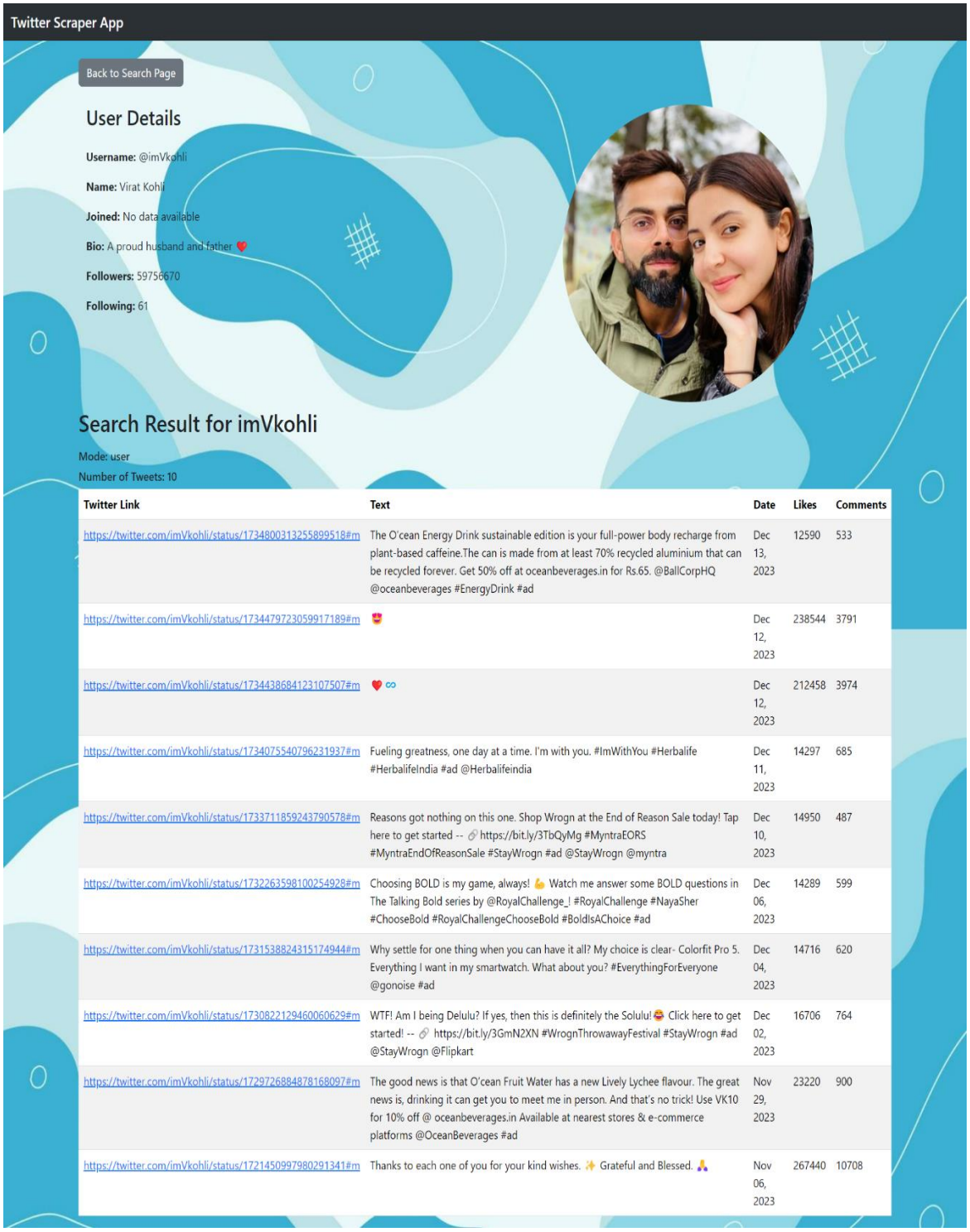
1. User entering the data into the form




2. Wait for tweets to load.

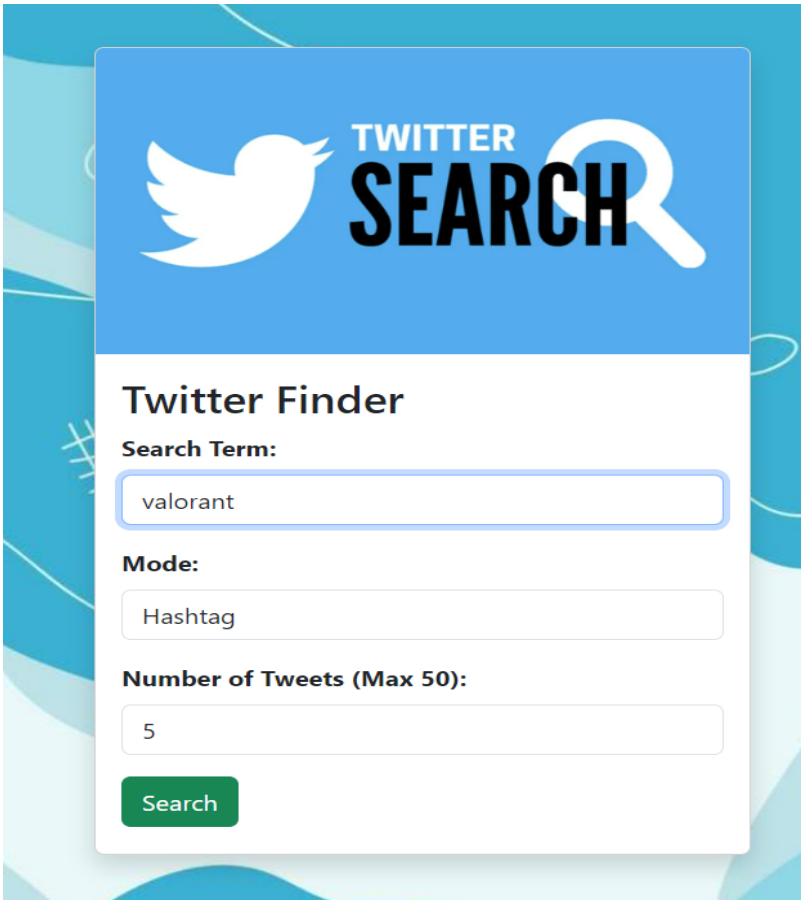


**Output in Mode : User**




| Twitter Link | Text | Date | Likes | Comments |
|---|---|---|---|---|
| https://twitter.com/imVkohli/status/1734800313255899518#m | The O'cean Energy Drink sustainable edition is your full-power body recharge from plant-based caffeine.The can is made from at least 70% recycled aluminium that can be recycled forever. Get 50% off at oceanbeverages.in for Rs.65. @BallCorpHQ @oceanbeverages #EnergyDrink #ad | Dec 13, 2023 | 12590 | 533 |
| https://twitter.com/imVkohli/status/1734479723059917189#m | 😍 | Dec 12, 2023 | 238544 | 3791 |
| https://twitter.com/imVkohli/status/1734438684123107507#m | ❤️ ∞ | Dec 12, 2023 | 212458 | 3974 |
| https://twitter.com/imVkohli/status/1734075540796231937#m | Fueling greatness, one day at a time. I'm with you. #ImWithYou #Herbalife #HerbalifeIndia #ad @Herbalifeindia | Dec 11, 2023 | 14297 | 685 |
| https://twitter.com/imVkohli/status/1733711859243790578#m | Reasons got nothing on this one. Shop Wrogn at the End of Reason Sale today! Tap here to get started -- 🔗 https://bit.ly/3TbQyMg #MyntraEORS #MyntraEndOfReasonSale #StayWrogn #ad @StayWrogn @myntra | Dec 10, 2023 | 14950 | 487 |
| https://twitter.com/imVkohli/status/1732263598100254928#m | Choosing BOLD is my game, always! 💪 Watch me answer some BOLD questions in The Talking Bold series by @RoyalChallenge_! #RoyalChallenge #NayaSher #ChooseBold #RoyalChallengeChooseBold #BoldIsAChoice #ad | Dec 06, 2023 | 14289 | 599 |
| https://twitter.com/imVkohli/status/1731538824315174944#m | Why settle for one thing when you can have it all? My choice is clear- Colorfit Pro 5. Everything I want in my smartwatch. What about you? #EverythingForEveryone @gonoise #ad | Dec 04, 2023 | 14716 | 620 |
| https://twitter.com/imVkohli/status/1730822129460060629#m | WTF! Am I being Delulu? If yes, then this is definitely the Solulu!😂 Click here to get started! -- 🔗 https://bit.ly/3GmN2XN #WrognThrowawayFestival #StayWrogn #ad @StayWrogn @Flipkart | Dec 02, 2023 | 16706 | 764 |
| https://twitter.com/imVkohli/status/1729726884878168097#m | The good news is that O'cean Fruit Water has a new Lively Lychee flavour. The great news is, drinking it can get you to meet me in person. And that's no trick! Use VK10 for 10% off @ oceanbeverages.in Available at nearest stores & e-commerce platforms @OceanBeverages #ad | Nov 29, 2023 | 23220 | 900 |
| https://twitter.com/imVkohli/status/1721450997980291341#m | Thanks to each one of you for your kind wishes. ✨ Grateful and Blessed. 🙏 | Nov 06, 2023 | 267440 | 10708 |

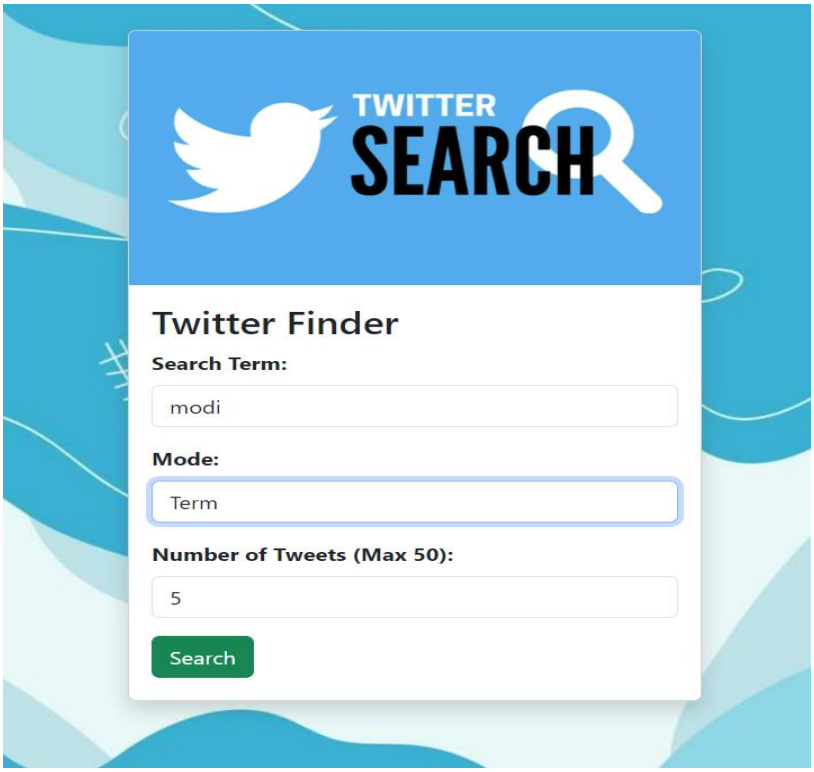
## Output 2 ( Mode : Hashtag ) –



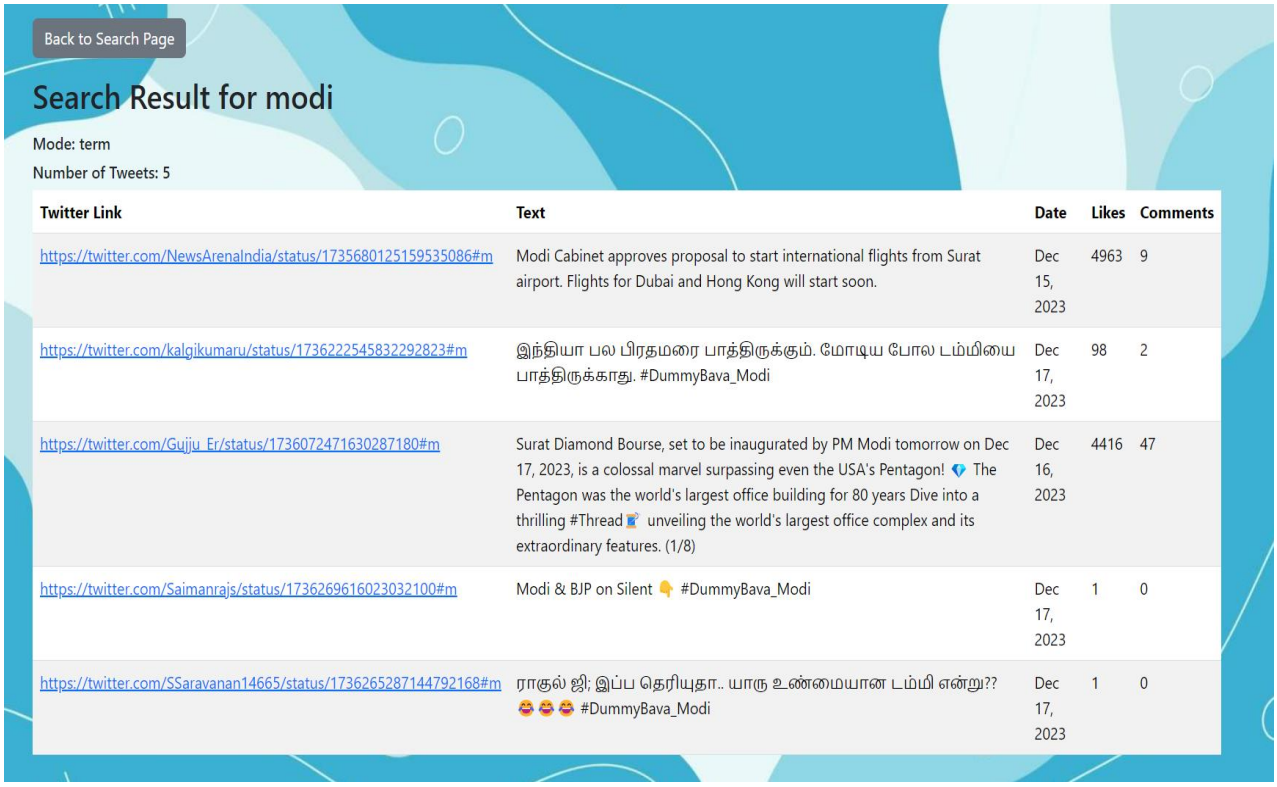

### Output in Mode : Hashtag


| Twitter Link | Text | Date | Likes | Comments |
|---|---|---|---|---|
| https://twitter.com/Spliinkles/status/1540055554282094592#m | Girls on my mind pt.2 \| #Valorant #Valorantart #Lightningstorm | Jun 23, 2022 | 5869 | 31 |
| https://twitter.com/Z3r0Tw0_ZT/status/1736268234591191418#m | Missed my ace #valorant #valorantclips Watch the full stream on YouTube and Twitch. Link in my profile.... | Dec 17, 2023 | 0 | 0 |
| https://twitter.com/is_min432/status/1736268210473943054#m | 勝利後味方を噛むスラッシュ #VALORANT #Gekko | Dec 17, 2023 | 0 | 0 |
| https://twitter.com/NAO09348850/status/1736048563107582274#m | なんでなん？？？ #VALORANT | Dec 16, 2023 | 20 | 1 |
| https://twitter.com/Rynies2/status/1736250004984258686#m | だいやいった まじでうれしい！！いいね&RTおねがいします！ #valorant #拡散希望 #valorant募集 | Dec 17, 2023 | 5 | 1 |

## Output 3 (Mode : Term )




### Output in Mode : Term




| Twitter Link | Text | Date | Likes | Comments |
|---|---|---|---|---|
| https://twitter.com/NewsArenaIndia/status/1735680125159535086#m | Modi Cabinet approves proposal to start international flights from Surat airport. Flights for Dubai and Hong Kong will start soon. | Dec 15, 2023 | 4963 | 9 |
| https://twitter.com/kalgikumaru/status/1736222545832292823#m | இந்தியா பல பிரதமரை பாத்திருக்கும். மோடிய போல டம்மியை பாத்திருக்காது. #DummyBava_Modi | Dec 17, 2023 | 98 | 2 |
| https://twitter.com/Gujju_Er/status/1736072471630287180#m | Surat Diamond Bourse, set to be inaugurated by PM Modi tomorrow on Dec 17, 2023, is a colossal marvel surpassing even the USA's Pentagon! 💎 The Pentagon was the world's largest office building for 80 years Dive into a thrilling #Thread 📝 unveiling the world's largest office complex and its extraordinary features. (1/8) | Dec 16, 2023 | 4416 | 47 |
| https://twitter.com/Saimanrajs/status/1736269616023032100#m | Modi & BJP on Silent 👇 #DummyBava_Modi | Dec 17, 2023 | 1 | 0 |
| https://twitter.com/SSaravanan14665/status/1736265287144792168#m | ராகுல் ஜி; இப்ப தெரியுதா.. யாரு உண்மையான டம்மி என்று?? 😂😂😂 #DummyBava_Modi | Dec 17, 2023 | 1 | 0 |