

**Department of Computer Science & Engineering**

**B. P. Poddar Institute of Management & Technology**

PCA-2 PROJECT

IT WORKSHOP (PYTHON) – PCC-CS-393

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| DEBANJAN GANGULY | 11500122001 |
| AMAN SANTRA | 11500122009 |
| ARGHA MALLICK | 11500122014 |
| AMRITASH BANERJEE | 2422127 |

**Project Title – News Summarization System**

**using Google RSS News Feed API & Natural Language Processing**

****

**Project Overview**

The News Summarization System is an innovative machine learning-driven project designed to streamline and condense news articles fetched from the Google RSS News Feed API. The system is engineered to collect a vast array of current news articles from various sources, process their content, and present concise and easily digestible summaries to users. Leveraging machine learning techniques, natural language processing (NLP), and text summarization algorithms, the system's core functionality involves converting lengthy news articles into shorter, more coherent summaries.

**Key Components and Workflow**

**Google RSS News Feed API Integration**: The system interfaces with the Google RSS News Feed API, allowing it to retrieve a diverse range of news articles across multiple categories and sources. This integration ensures a constant inflow of up-to-date information.

**Data Preprocessing**: Upon retrieval of news articles, the data undergoes preprocessing, including text cleaning, tokenization, and filtering out redundant or irrelevant information. This step prepares the content for subsequent analysis.

**Natural Language Processing (NLP):** NLP techniques are applied to comprehend the context, semantics, and structure of the news articles. This includes tasks such as part-of-speech tagging, named entity recognition, and sentiment analysis to grasp the nuances of the text.

**Text Summarization Algorithms**: Employing machine learning-driven summarization algorithms such as extractive or abstractive methods, the system condenses the articles into brief, coherent summaries. Extractive methods may involve ranking and selecting key sentences, while abstractive methods might involve generating new sentences based on the content.

**User Interface and Different Routes**: The system provides a user-friendly interface through different routes or paths, allowing users to access summarized news in various categories or topics. This may include options to explore summarized news by subject, date, source, or user preferences. Also, user can search for any particular topics in the search bar to view the related news articles.

**Output and User Interaction:**

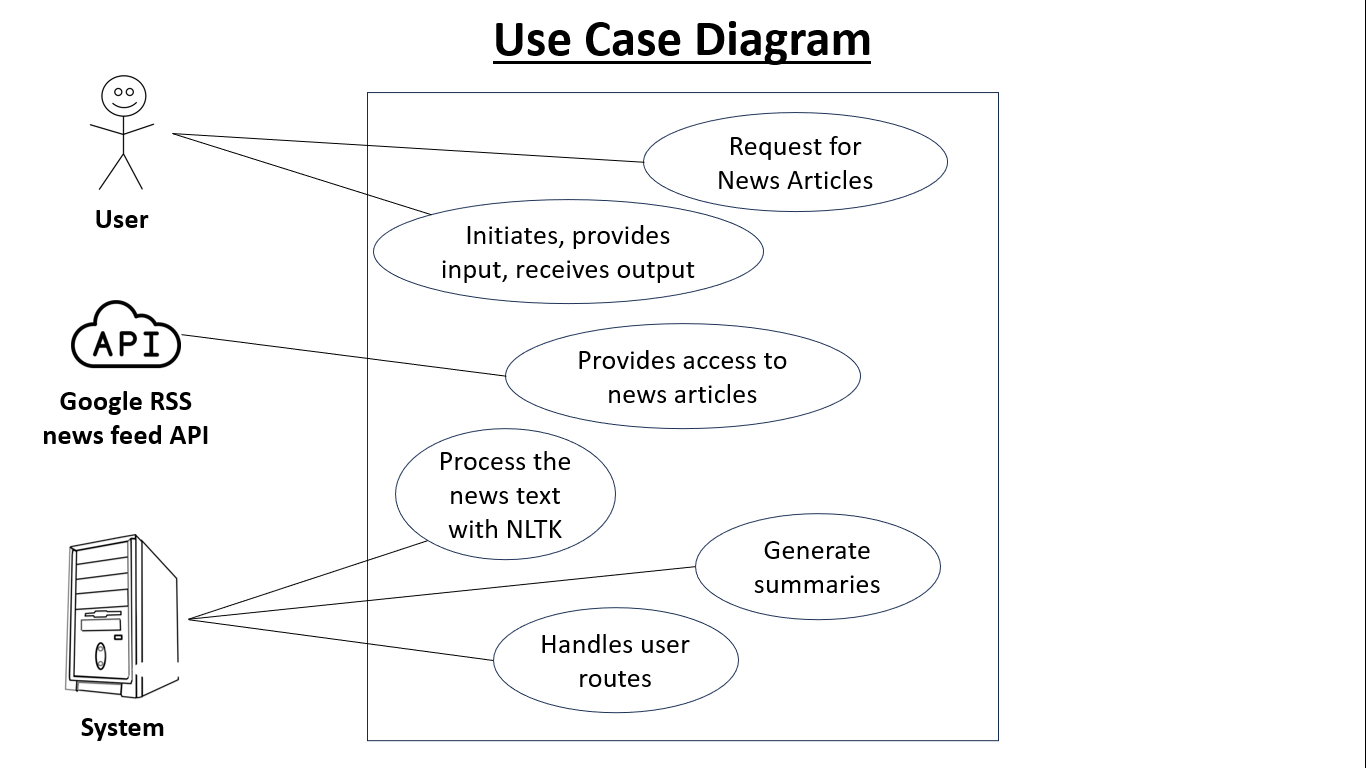
The summarized news articles are then displayed to the user through the chosen routes, enabling quick and efficient consumption of essential information. Users can access a wide array of summarized news articles, saving time while staying informed about current events.

**Benefits and Future Prospects**

This project aims to offer users a convenient way to stay updated on current events without investing extensive time in reading lengthy articles. It facilitates quick access to crucial information, making it beneficial for individuals with time constraints or a need for rapid news consumption.

Future iterations of the project might incorporate improvements in summarization techniques, personalization features, user feedback integration, and enhanced user interfaces for an even more tailored and engaging experience.

This project holds significant potential in providing users with a more efficient and accessible method for staying informed about the latest news, and it showcases the powerful applications of machine learning in content summarization.

****

**SOURCE CODE**

**# app.py**

from flask import Flask, render\_template, request

from bs4 import BeautifulSoup as soup

from urllib.request import urlopen

from newspaper import Article

import nltk

# nltk.download('punkt')

def fetch\_news\_search\_topic(topic):

    topic = topic.replace(" ", "%20")

    site = 'https://news.google.com/rss/search?q={}'.format(topic)

    op = urlopen(site)  # Open that site

    rd = op.read()  # read data from site

    op.close()  # close the object

    sp\_page = soup(rd, 'xml')  # scrapping data from site

    news\_list = sp\_page.find\_all('item')  # finding news

    return news\_list

def fetch\_top\_news():

    site = 'https://news.google.com/news/rss'

    op = urlopen(site)  # Open that site

    rd = op.read()  # read data from site

    op.close()  # close the object

    sp\_page = soup(rd, 'xml')  # scrapping data from site

    news\_list = sp\_page.find\_all('item')  # finding news

    return news\_list

def fetch\_category\_news(topic):

    site = 'https://news.google.com/news/rss/headlines/section/topic/{}'.format(topic.upper())

    op = urlopen(site)  # Open that site

    rd = op.read()  # read data from site

    op.close()  # close the object

    sp\_page = soup(rd, 'xml')  # scrapping data from site

    news\_list = sp\_page.find\_all('item')  # finding news

    return news\_list

def display\_news(list\_of\_news, news\_quantity):

    c = 1

    response = []

    for news in list\_of\_news:

        news\_data = Article(news.link.text)

        try:

            news\_data.download()

            news\_data.parse()

            news\_data.nlp()

        except Exception as e:

            print("Error:", e)

        news\_id = str(c) + 'NewsBits'

        news\_title = news.title.text

        news\_summery = news\_data.summary

        news\_source = news.source.text

        news\_link = news.link.text

        news\_date = news.pubDate.text

        news\_image = news\_data.top\_image

        data\_dict = {}

        data\_dict["news\_id"] = news\_id

        data\_dict["news\_title"] = news\_title

        data\_dict["news\_image"] = news\_image

        data\_dict["news\_summery"] = news\_summery

        data\_dict["news\_source"] = news\_source

        data\_dict["news\_link"] = news\_link

        data\_dict["news\_date"] = news\_date

        response.append(data\_dict)

        print(f"Fetched News {c}")

        if c >= news\_quantity:

            break

        c += 1

    return response

app = Flask(\_\_name\_\_)

@app.errorhandler(404)

def not\_found(e):

    print(e)

    return render\_template('404.html')

default\_news\_quantity = 2

@app.route('/', methods=["GET","POST"])

def home\_page():

    if request.method == "GET":

        print("GET '/'")

        news\_list = fetch\_top\_news()

        response = display\_news(list\_of\_news=news\_list,

                                news\_quantity=default\_news\_quantity)

        return render\_template('home.html', response=response, topic="Top Trending", news\_quantity=default\_news\_quantity, request\_url="/")

    if request.method == "POST":

        load\_more\_range = int(request.form.get('load\_more\_range'))

        print(f"Loading more on '/' with news quantity: {load\_more\_range}")

        news\_list = fetch\_top\_news()

        response = display\_news(list\_of\_news=news\_list,

                                news\_quantity=load\_more\_range)

        return render\_template('home.html', response=response, topic="Top Trending", news\_quantity=load\_more\_range, request\_url="/")

@app.route('/topics/<string:topic>', methods=["GET", "POST"])

def search\_by\_topics(topic):

    news\_list = fetch\_category\_news(topic)

    if request.method == "GET":

        print(f"GET '/topics/{topic}'")

        response = display\_news(list\_of\_news=news\_list,

                                news\_quantity=default\_news\_quantity)

        return render\_template('home.html', response=response, topic=topic.upper(), news\_quantity=default\_news\_quantity, request\_url="/topics/"+topic)

    if request.method == "POST":

        load\_more\_range = int(request.form.get('load\_more\_range'))

        print(f"Loading more on '/topics/{topic}' with news quantity: {load\_more\_range}")

        response = display\_news(list\_of\_news=news\_list,

                                news\_quantity=load\_more\_range)

        return render\_template('home.html', response=response, topic=topic.upper(), news\_quantity=load\_more\_range, request\_url="/topics/"+topic)

@app.route('/search', methods=["GET","POST"])

def search():

    if request.method == "GET":

        print("GET '/search'")

        return render\_template('home.html', request\_url="/search")

    if request.method == "POST":

        try:

            load\_more\_range = int(request.form.get('load\_more\_range'))

        except:

            load\_more\_range = default\_news\_quantity

        topic = request.form.get('topic')

        print(f"Searching for {topic} with news quantity: {load\_more\_range}")

        news\_list = fetch\_news\_search\_topic(topic)

        response = display\_news(list\_of\_news=news\_list,

                                news\_quantity=load\_more\_range)

        return render\_template('home.html', response=response, topic=topic, news\_quantity=load\_more\_range, request\_url="/search")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host='0.0.0.0', port=3000,debug=True)

**# templates/layout.html**

<!doctype html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

    <title>News Summerization System</title>

    <link type="image/png" sizes="32x32" rel="icon" href="{{ url\_for('static', filename='/images/favicon.png') }}">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="{{ url\_for('static', filename='/css/style.css') }}">

    <link rel="stylesheet" href="{{ url\_for('static', filename='/css/scrollbar.css') }}">

</head>

<body>

    <nav class="navbar navbar-dark bg-dark navbar-expand-lg" style="background-color: #735d7e !important;">

        <div class="container-fluid">

            <a class="navbar-brand" href="/">NewsBits</a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"

                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"

                aria-label="Toggle navigation">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">

                <ul class="navbar-nav ms-auto mb-2 mb-lg-0" style="padding-right: 10px;">

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/">HOME</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/world">WORLD</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/nation">NATION</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/business">BUSINESS</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/technology">TECH</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/entertainment">ENTERTAINMENT</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/sports">SPORTS</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link active" aria-current="page" href="/topics/science">SCIENCE</a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <main class="bg-dark min-h-100 pb-5">

        <div class="container">

            <img class="w-100 mt-3" src="{{ url\_for('static', filename='/images/newspaper.png') }}" alt="" srcset=""

                style="border-radius: 50%;">

            <h1 class="text-light text-center">Welcome to NewsBits📰 Read the Summarized News Articles</h1>

            <form action="/search" method="POST" class="p-3">

                <div class="form-group mb-3 form-content">

                    {% if topic %}

                    <input value="{{topic}}" name="topic" type="text" class="form-control" id="topic"

                        placeholder="Search Topics" required>

                    {% else %}

                    <input name="topic" type="text" class="form-control" id="topic" placeholder="Search Topics"

                        required>

                    {% endif %}

                    <button type="submit" class="btn btn-primary"><i class="fa fa-search"></i></button>

                </div>

            </form>

        </div>

        {% block body %}

        {% endblock body %}

    </main>

    <footer>

        <p>News Summerization System &copy; Made with Flask by Mechine Learning Integrity</p>

    </footer>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

        crossorigin="anonymous"></script>

</body>

</html>

**# templates/home.html**

{% extends 'layout.html' %}

{% block body %}

{% if response %}

<h2 class="text-center text-light pb-3">Showing News for <strong>{{topic}}</strong></h2>

<div class="container home text-light">

    {% for news in response %}

    <div class="news-card">

        <img src="{{news['news\_image']}}" alt="" srcset="" loading="lazy">

        <div class="accordion" id="accordionExample{{news['news\_id']}}">

            <div class="accordion-item">

                <h2 class="accordion-header">

                    <button class="accordion-button text-light" type="button" data-bs-toggle="collapse"

                        data-bs-target="#collapseOne{{news['news\_id']}}" aria-expanded="true"

                        aria-controls="collapseOne{{news['news\_id']}}"

                        style="background-color: #735d7e;">{{news['news\_title']}}

                    </button>

                </h2>

                <div id="collapseOne{{news['news\_id']}}" class="accordion-collapse collapse show"

                    data-bs-parent="#accordionExample{{news['news\_id']}}">

                    <div class="accordion-body bg-dark">

                        {{news['news\_summery']}}

                        <br>

                        <p class="font-weight-bold"><a class="text-warning font-weight-bold" target="\_blank"

                                href="{{news['news\_link']}}">Read

                                full

                                article on

                                {{news['news\_source']}}</a></p>

                    </div>

                </div>

            </div>

        </div>

        <p class="bg-success text-center">Published Date {{news['news\_date']}}</p>

    </div>

    {% endfor %}

</div>

<div class="container mt-4">

    <form action="{{request\_url}}" method="POST">

        <div class="load\_more">

            <input value="{{topic}}" name="topic" type="text" class="form-control" id="topic"

                placeholder="Search Topics" hidden>

            <label class="btn btn-light">Number of Articles ({{news\_quantity}})</label>

            <button type="submit" class="btn btn-primary">Go</button>

        </div>

        <input class="w-100" type="range" name="load\_more\_range" id="load\_more\_range" step="1" min="1" max="25"

            value="{{news\_quantity}}">

        <div class="load\_more text-light">

            <span>01</span><span>05</span><span>10</span><span>15</span><span>20</span><span>25</span>

        </div>

    </form>

</div>

{% else %}

<div class="container">

    <a href="/"><img class="w-100 rounded" src="{{ url\_for('static', filename='/images/403.png') }}"

            alt="ERROR 403 : access denied or http request failed"></a>

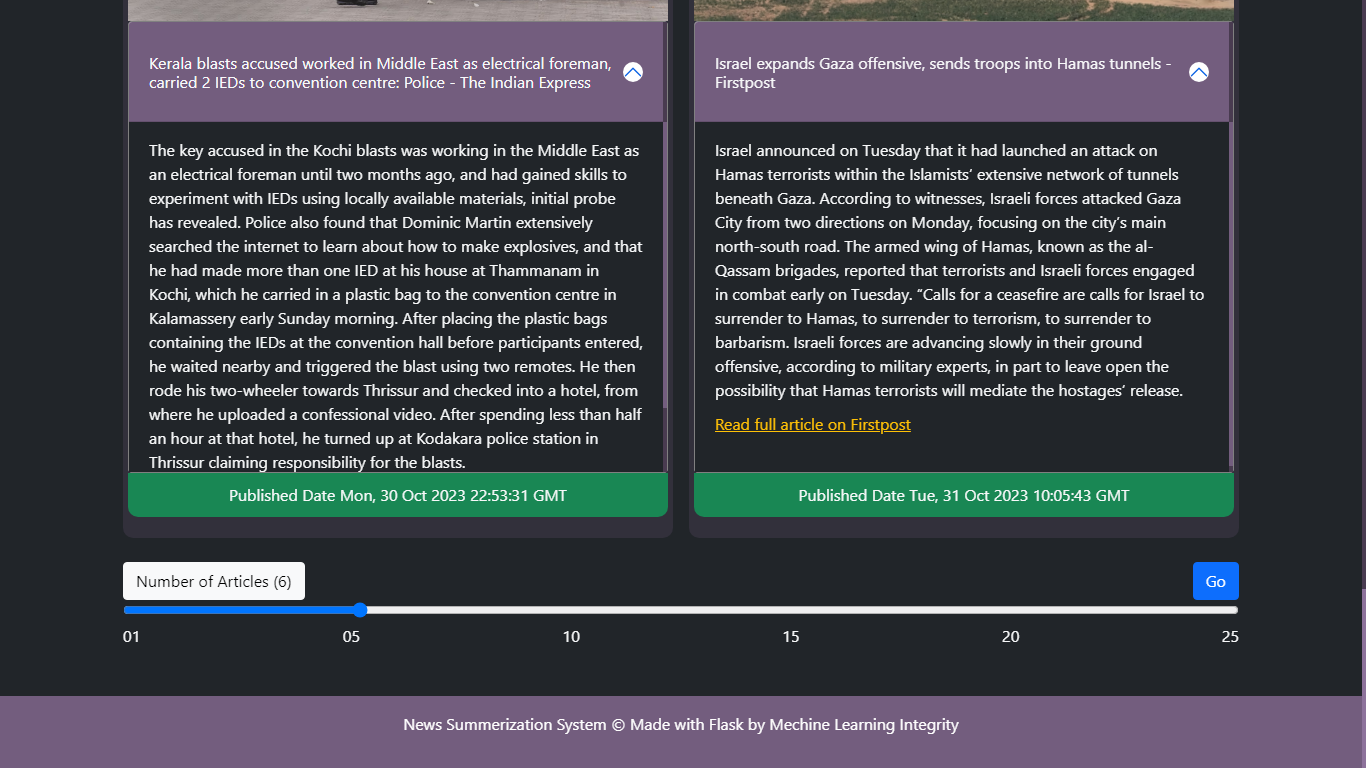
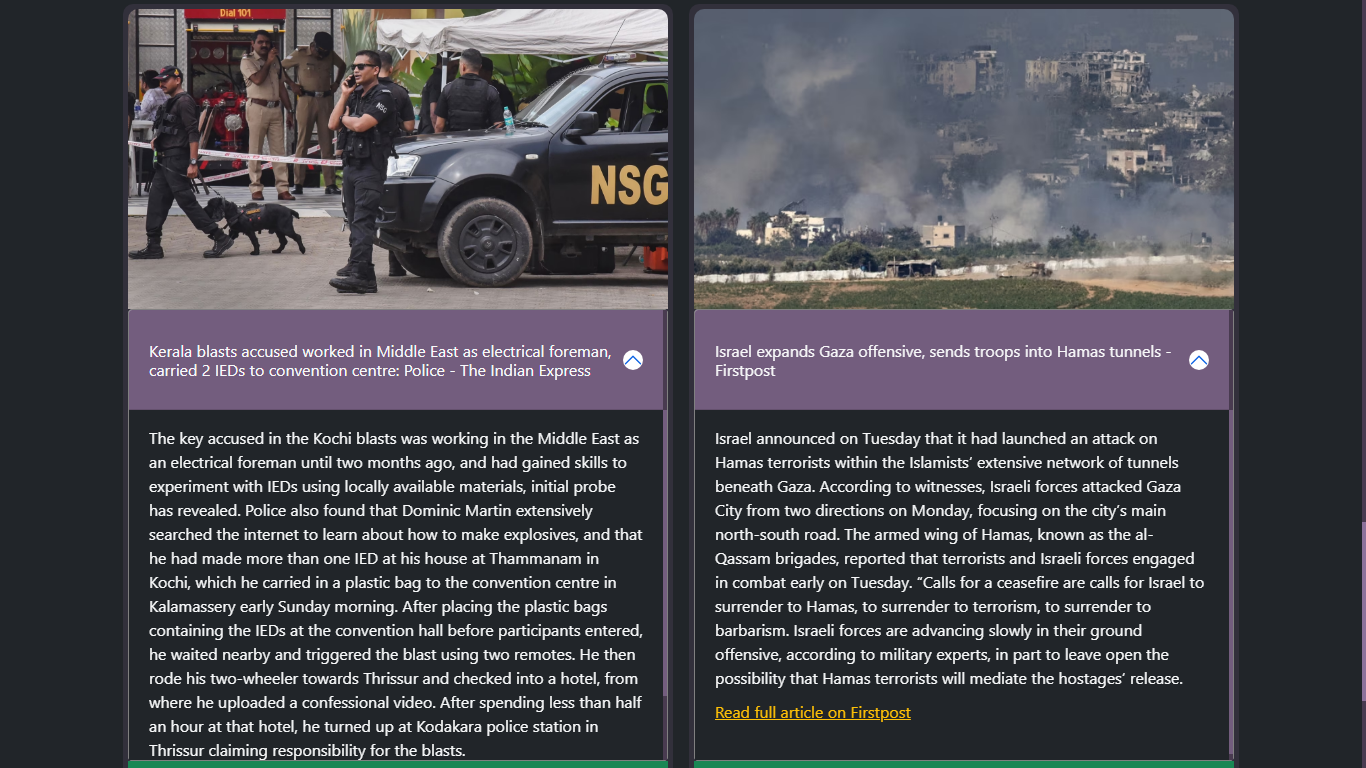
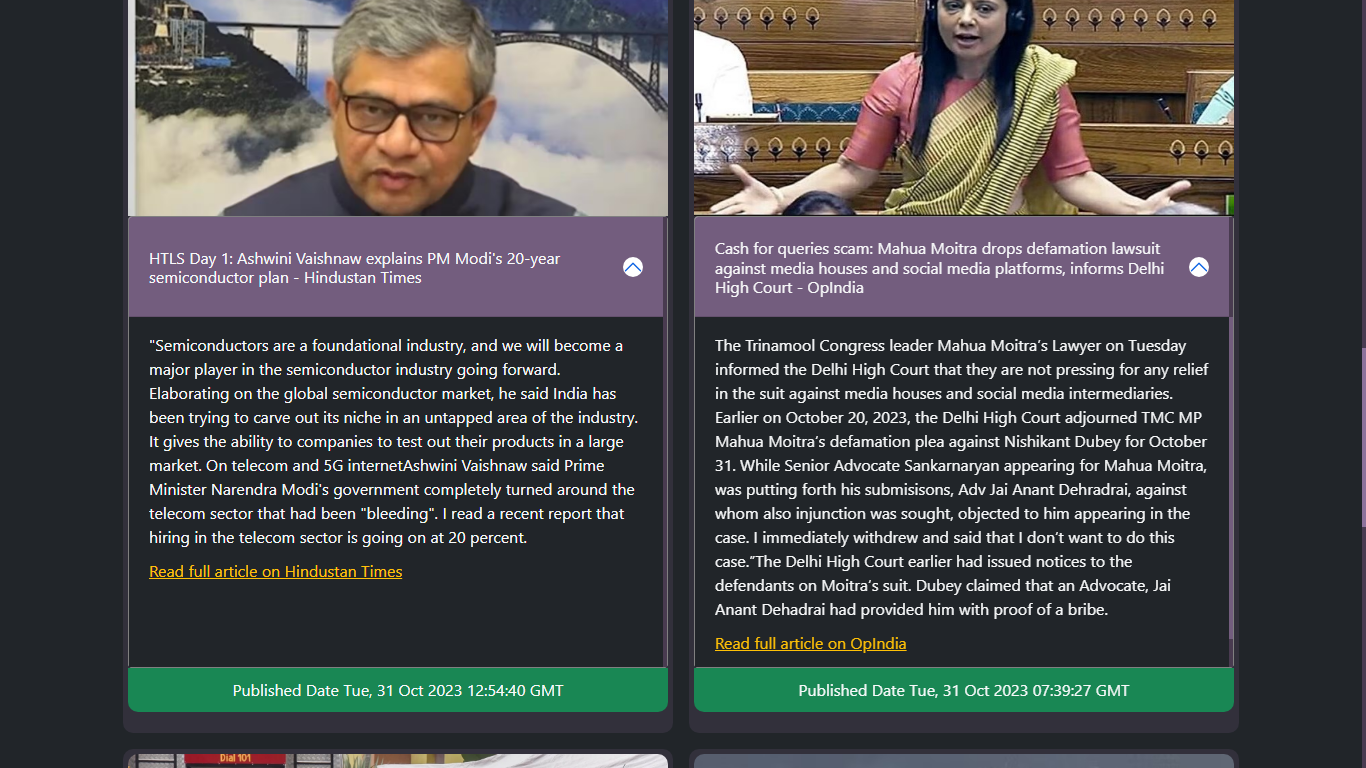
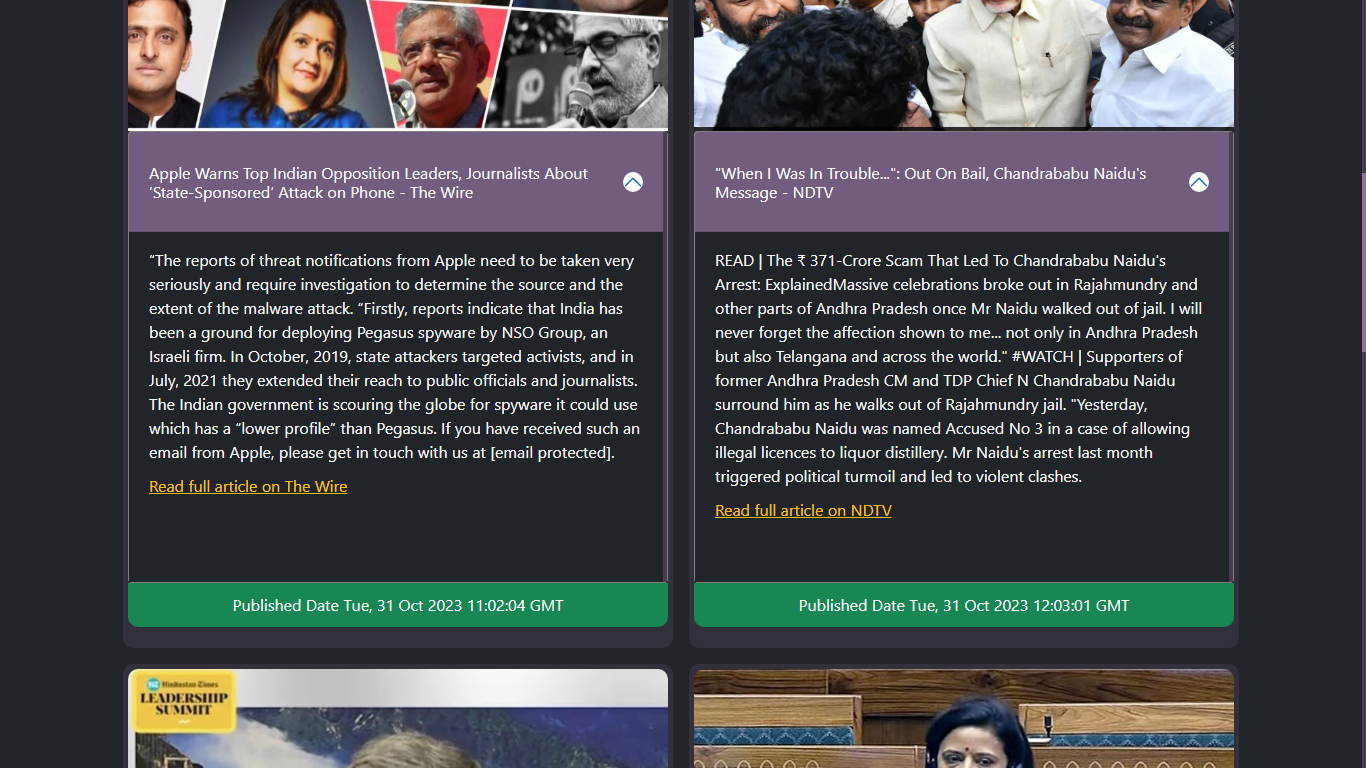
</div>

{% endif %}

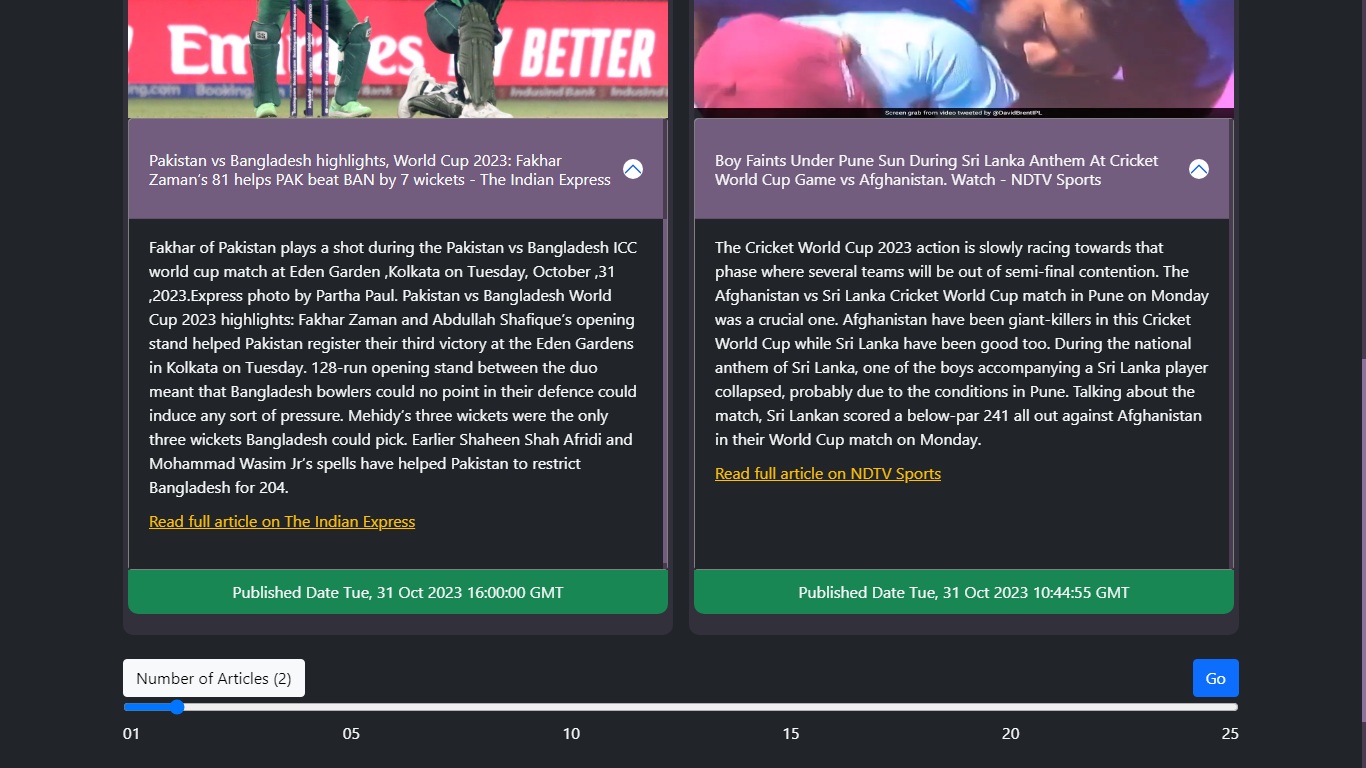
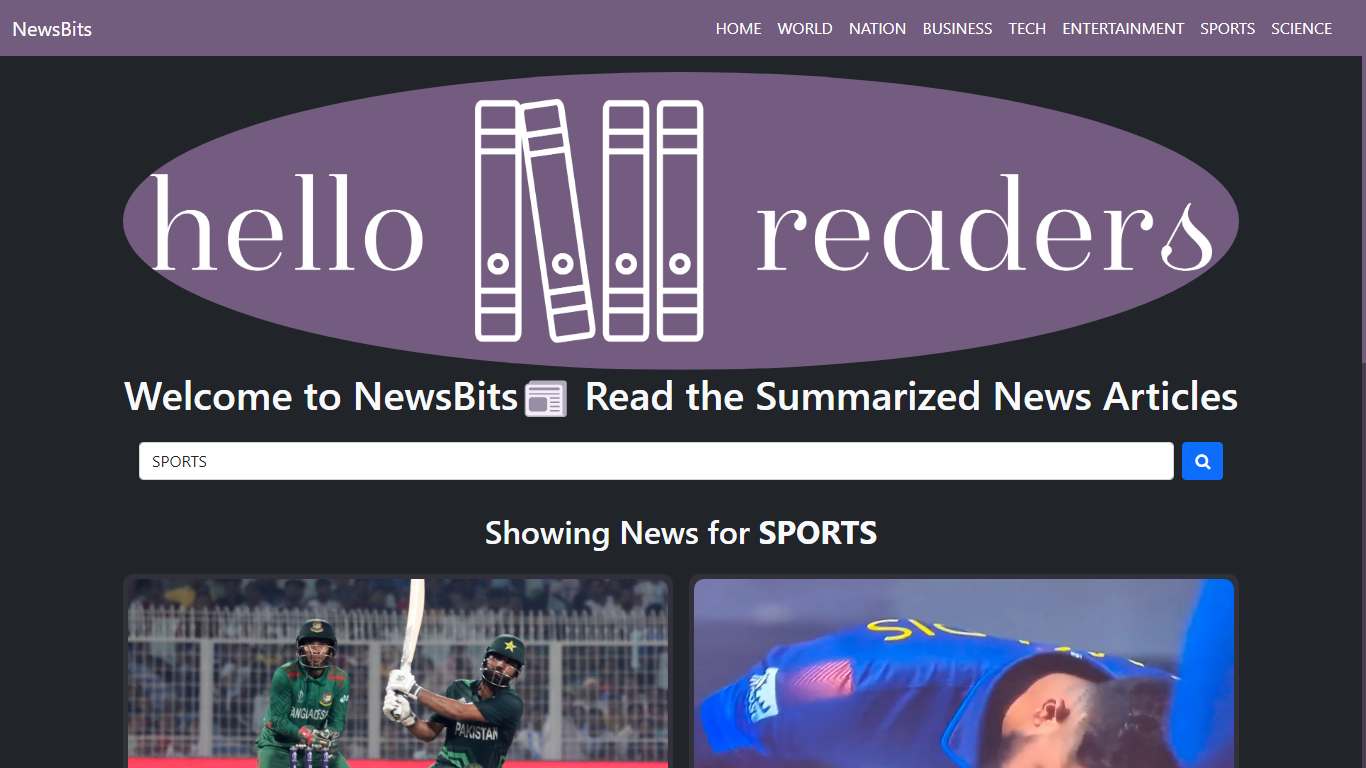
{% endblock %}

**OUTPUT SCREEN**

End point: Homepage( ‘/’ )

****

End point: Topics ( ‘/topics/sports’ )



**Other similar endpoints are:**

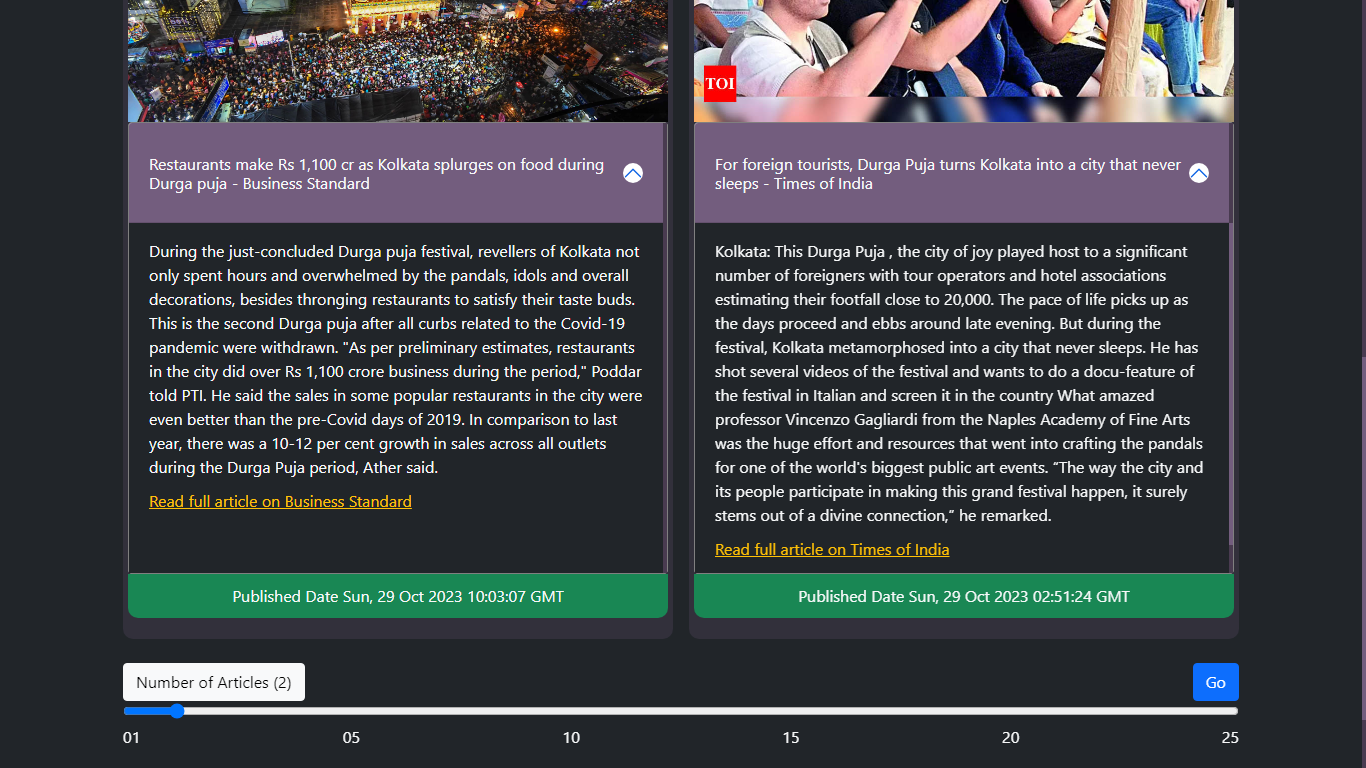
/topics/world /topics/entertainment

/topics/nation /topics/science

/topics/business

/topics/tech

**Search Endpoint:** /search

****

Link to the deployment (demo):

<https://newsbits.onrender.com/>