

NEWS TRACKER APPLICATION

IBM PROJECT

TEAM_ID:PNT2022TMID12184

Submitted by

920419UCSE001 VIGNESHKUMAR S

920419UCSE068 GANESHGURU A

920419UCSE003 NAVEEN R

920419UCSE100 MOHAN RAM T

in partial fulfillment for the award of

the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KAMARAJ COLLEGE OF ENGINEERING

AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

NOV 2022

TABLE OF FIGURES

1. INTRODUCTION
 - 1.1 Project Overview
 - 1.2 Purpose
2. LITERATURE SURVEY
 - 2.1 Existing problem
 - 2.2 References
 - 2.3 Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION
 - 3.1 Empathy Map Canvas
 - 3.2 Ideation & Brainstorming
 - 3.3 Proposed Solution
 - 3.4 Problem Solution fit
4. REQUIREMENT ANALYSIS
 - 4.1 Functional requirement
 - 4.2 Non-Functional requirements
5. PROJECT DESIGN
 - 5.1 Data Flow Diagrams
 - 5.2 Solution & Technical Architecture
 - 5.3 User Stories
6. PROJECT PLANNING & SCHEDULING
 - 6.1 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
7. CODING & SOLUTIONING
 - 7.1 Feature
8. TESTING
 - 8.1 Test Cases
 - 8.2 User Acceptance Testing
9. RESULTS
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX(Source Code, GitHub & Project Demo Link)

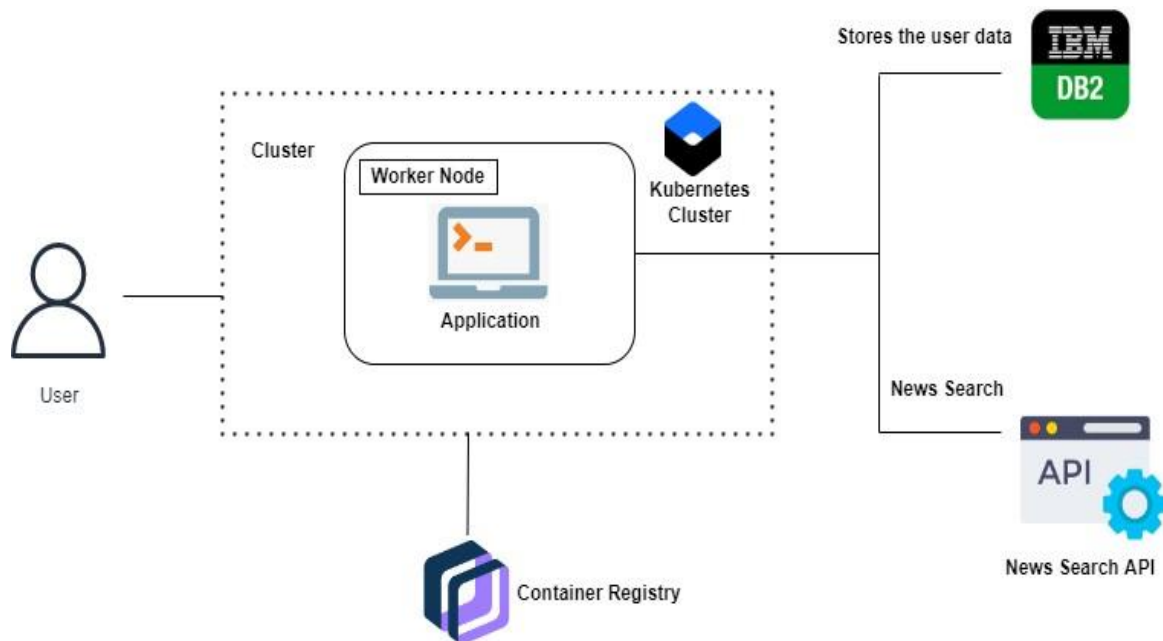
1. INTRODUCTION :

Nowadays, the news media are subject to the social scrutiny because of the lack of credibility, the manipulative media, the misinformation campaigns, and the propagation of fake news point out that the main difference between fake news and true news relies on lack editorial norms and processes that ensure the accuracy and credibility of the information. Thus, to arrange a way that allows guaranteeing these editorial processes (or at least part of them) can suppose a big step in the fight against the above-mentioned issues. To track the evolution of the news reports and the relevant data and information it contains as they change over time, and therefore to trace how the related news evolves, constitute other instruments to face the previous issues. This is not only useful for end readers but for fact-checking agencies and those tools that perform automatic indexing and extraction of relevant information of news. Once the information has been verified and fact-checked, these tools need a way to guarantee that the extracted data have not changed.

1.1 Project Overview

News Tracker Application

Technical Architecture:



Project Workflow:

- The user interacts with the application.
- Registers by giving the details.
- Integrate the application with news APIs and store the data in the database.
- The database will have all the details and the user can search the news by using a search bar.

1.2 Purpose

As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.... as on official websites.

2. LITERATURE SURVEY**2.1 Existing problem**

News is one of the primary source of gaining information about the actions and events that happen all around. It may be an event that happened in the past, happening now or going to happen in the future. In the present days where there is a rapid increase in the development and adaptability of technologies throughout all the demographic of people, it is necessary to provide news in such a way that it is interconnected with the current technological trends. As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.... as on official websites

2.2 References

S.No	Paper Title	Author(s)	Month/ Year	Method/ Implementation Technique(s)	Resource Link
1	Exploring mobile news reading interactions for news app personalization	Marios Constantinides, John Dowell, David Johson, Sylvain Malacria	August, 2015	1. Identification of news reader types 2. Interaction logging and classification study 3. Deployment and data collection 4. Predicting News readertypes 5. Adaptive UI	https://www.researchgate.net/publication/299870645_Exploring_mobile_news_reading_interactions_for_news_app_personalisation
2	Detection and Tracking in News Articles	Sagar Patel, Sanket Suthar, Sandip Patel, Neha Patel	March, 2015	1. Preprocessing 2. Tokenization 3. Stemming/L emmization 4. Vector SpaceModel 5. Topic tracking	https://www.researchgate.net/publication/315657099_Topic_Detection_and_Tracking_in_News_Articles
3	Following the Fed with a News Tracker	Michael William McCracken	January, 2012	The paper is not a technical paper but is essentially a statistical paper on how should one conclude whether the data have come in stronger, weaker or as expected. This is based on the CitiGroup U.S Economic Surprise Index.	https://www.researchgate.net/publication/227438253_Following_the_Fed_with_a_News_Tracker
4	An End-to-end Weaklysupervised News Aggregation Framework	Xijin Tang, Xiaohui Huang	June, 2022	The framework combines Snorkelbased weaklysupervised classification, Latent Dirichlet Allocation (LDA) topic modeling, and topic signal detection model to classify and aggregate unlabeled	https://www.researchgate.net/publication/361087328_An_End-to-end_Weakly-supervised_News_Aggregation_Framework

				news texts and ultimately generate visualized results containing news categories, news topics, and temporal topic relationships. This paper uses constructed knowledge thesaurus and the Snorkel method to weakly supervise the classification of unlabeled news with no manual tagging. Subsequently, we utilize LDA to generate the topics and obtain the signal value of each topic based on the topic signal detection function. Finally, we establish the temporal topic relationships and get the visualized results of news aggregation.	
--	--	--	--	---	--

2.3 Problem Statement Definition

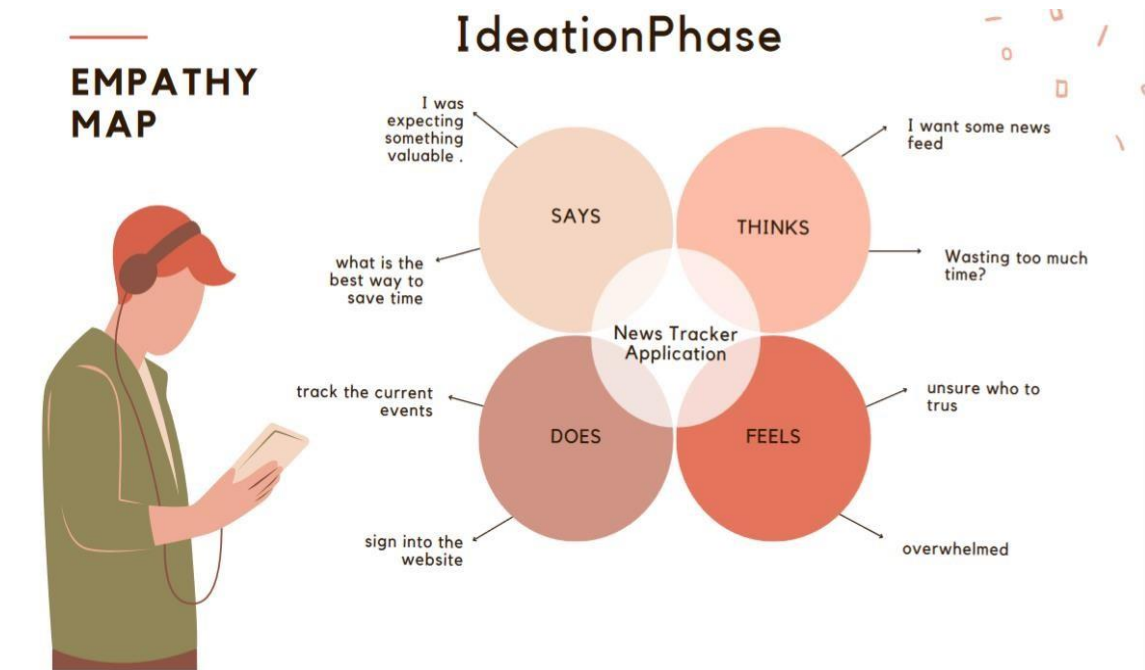
Mr. John is a 52 years old man, who frequently reads news, he wishes to read the pertinent market news and needs to get help with queries about indices, commodities, currencies, future rates, bond, etc... Through News Tracker Application.

Who does the problem affect?	Person who searches for authentic news.
What are the boundaries of the problem?	People with lack of internet facility and facing the difficulty in operating the software.
What is the issue?	As our lives are very busy now-a-days, we often feel we need more than 24 hrs. A day to come up with everything to be updated ourself via news in our schedule. Well, that's not possible but reducing the time by using the conventional method of reading news can help at any time in any place to be updated.
When does the issue occur?	When we search for pertinent news.

Where does the issue occur?	This issue occurs for common people, business men.
Why is it important that we fix the problem?	It is important to read the relevant news and save time.
What solution to solve this issue?	This could be solved by collecting relevant information from the user through a news tracker application. App will provide a quick snap to save your time.
What methodology was used to the issue?	<ul style="list-style-type: none"> ➤ The User interacts with the application. ➤ Registers by giving the details. ➤ Integrate the application with News APIs and store the data in Database. ➤ The database will have all the details and the user can search the news by using a search bar.


3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas




3.2 Ideation & Brainstorming


Template





Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 **10 minutes** to prepare

 **1 hour** to collaborate

 **2-8 people** recommended

 [Share template feedback](#)



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes



Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.



Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.



Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

The user needs a way to get relevant news based on his choices so that the user does not have to spend a lot of time on searching news. News is filled with ads and spams annoys and irritates the user and affects the user experience. Market is full of news with all categories. Users are not interested in all the categories and will be more interested with their personal choices. Traditional way of tracking news is slow and obsolete, user needs a new innovative application to track news with personal based choices. Lack of quick updates for the day results in staying behind in the trend in today's world and tracking the news regularly should be done.



Key rules of brainstorming

To run an smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP



You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

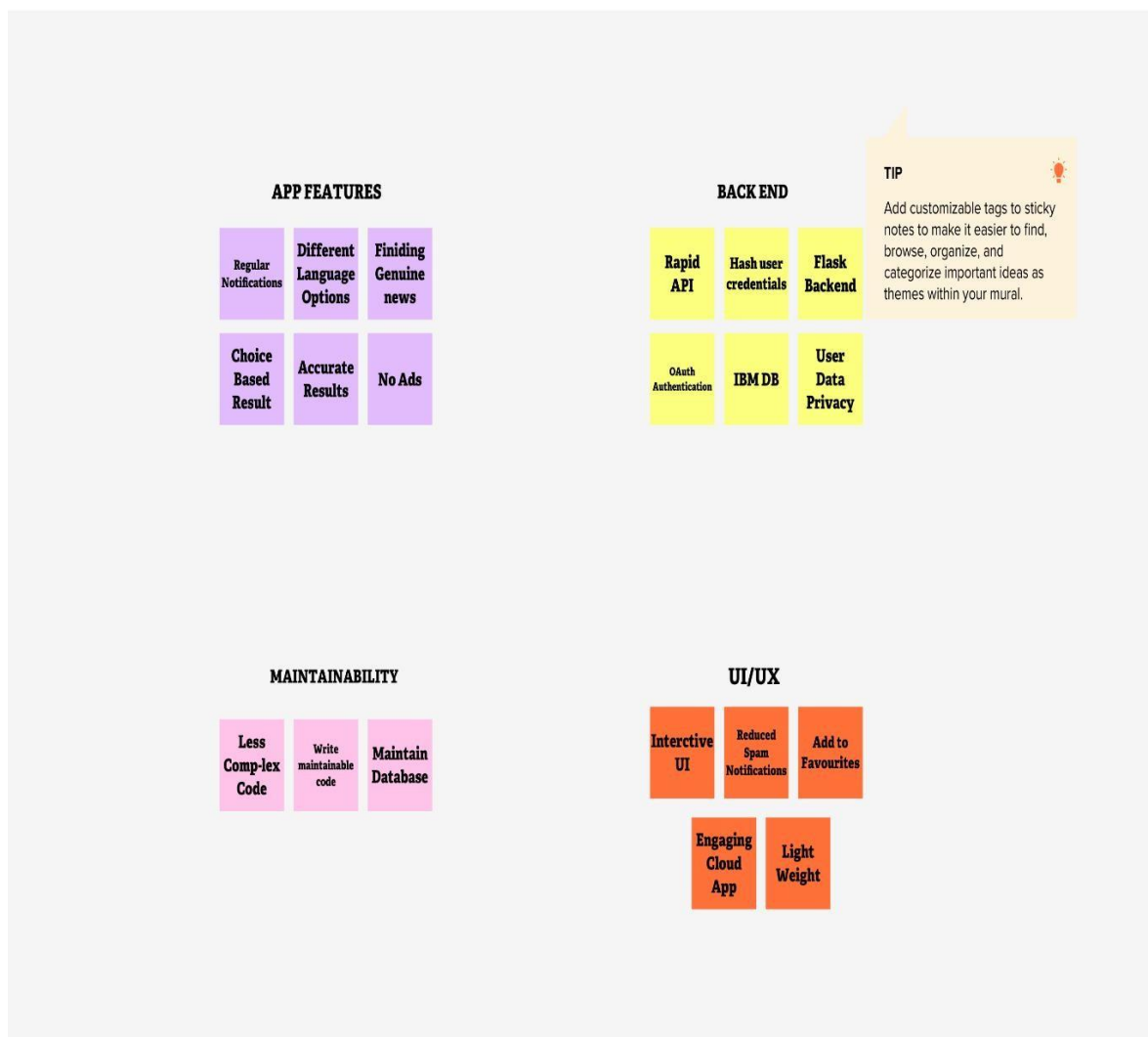


3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

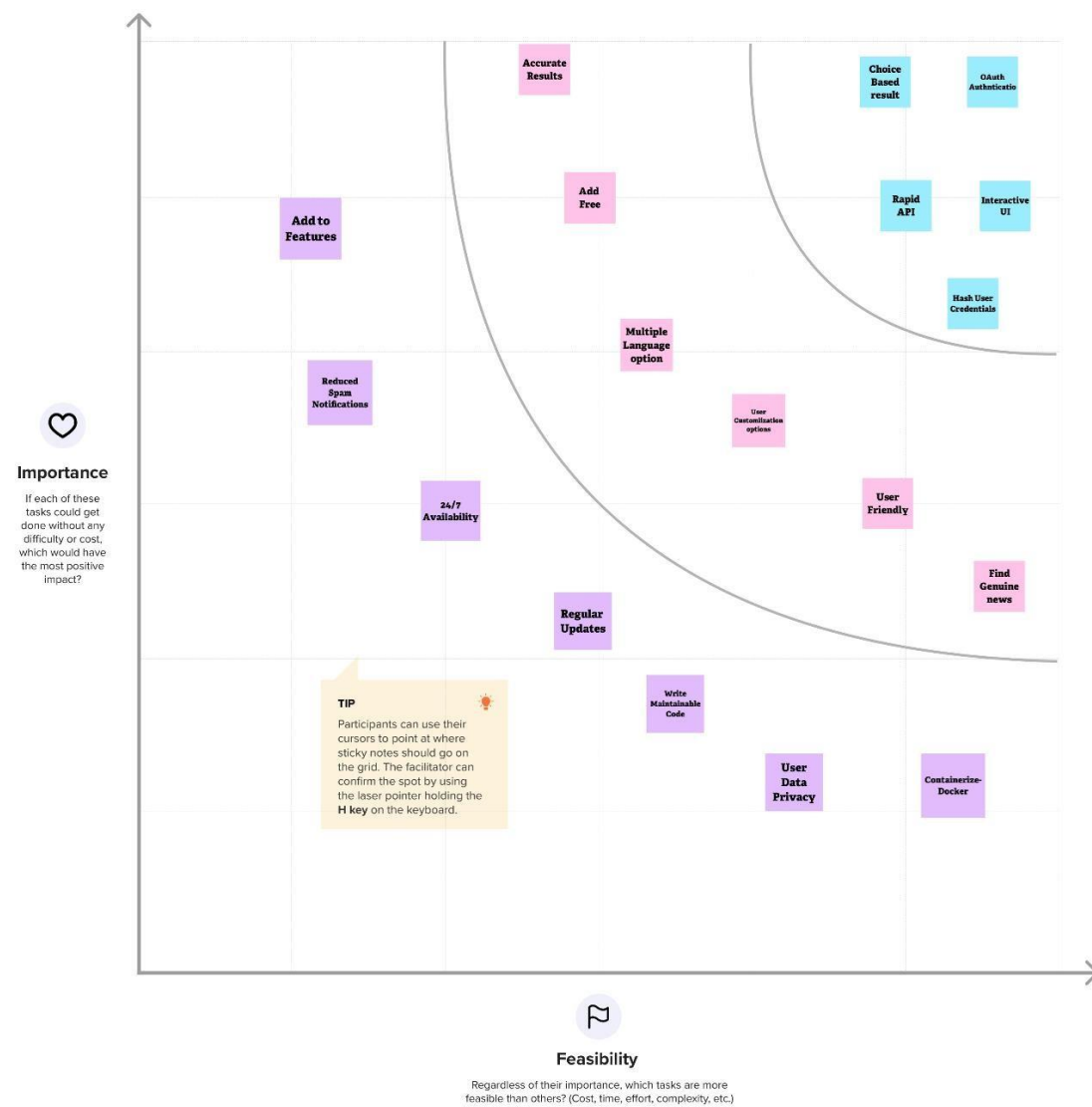


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes





After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons



Share the mural

Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.



Export the mural

Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward



Strategy blueprint

Define the components of a new idea or strategy.

[Open the template →](#)



Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

[Open the template →](#)



Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

[Open the template →](#)



[Share template feedback](#)

3.3 Proposed Solution

ABSTRACT

News Tracker is to collect a variety of sources of misinformation on social media and aggregate the stories published so that we could have a clearer sense of the different types of misinformation out there. We wanted to know how frequently it was published and what strategies and narratives were employed to engage audiences on social media. To do this, we built a database of misinformation sites, identified their social media pages, and found any related to that also shared the content. We used the social media API to collect all the stories posted to these pages as well as their engagement data — how many times users liked, shared, and commented on the stories. We continuously monitored this feed to identify patterns in and assess the substance of the posts. Fake news is a false piece of information. Any piece of Fake news can be created to deliberately misinform or deceive readers, promote a biased point of view, particular cause or agenda, and even for the entertainment. Fake news can be promoted by unauthenticated user ID, social media, printing of fake news in newspapers may be due to political pressure and many more. Spreading of fake news can cause discontent among people, riots, and even cause loss of trust between two people and even nations. It also has a huge possibility to exploit general public thinking in a completely different manner. News and media coverage gets hugely distorted due to initialization and spread of fake news.

PROPOSED SYSTEM

The proposed system we are using a hybrid support vector machine and Random forest model for training. We are combining the two machine learning techniques so that we can able to get the accuracy score of above 98% for large datasets also.

3.4 Problem Solution fit

Project Title: NEWS TRACKER APPLICATION

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMD26501

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS <p>The one who can't find time to read news paper</p> <p>People who just want to read relevant news.</p>	6. CUSTOMER CONSTRAINTS CC <p>Network issues while scrolling for news</p> <p>Responsivity of application</p> <p>Giving relevant news according to their search.</p>	5. AVAILABLE SOLUTIONS AS <p>They may choose newspaper or YouTube for news.</p> <p>Various news applications are available in various languages.</p>	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <p>They can't reschedule / give separate time to read news paper</p> <p>The problem while searching for the relevant news</p> <p>The particular news can't be searched in case of newspaper.</p> <p>The interests of the customer can't be managed rather than digital application.</p>	9. PROBLEM ROOT CAUSE RC <p>Newspapers give day-to- day news only.</p> <p>A day to cope with everything we have in our schedule</p>	7. BEHAVIOUR BE <p>People may use applications but, they have constraints</p> <p>People may surf the internet but, won't get the apt news.</p> <p>YouTube channels may be a solution but, they can't get all the news available.</p>	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS TR <p>Customer may install the app by seeing ads.</p> <p>Through friends' suggestion.</p>	10. YOUR SOLUTION SL <p>User should login with their details in the application in which we are providing with databases to manage their interest.</p> <p>Easy to use</p> <p>Various languages and regional news will be there.</p>	8. CHANNELS of BEHAVIOUR CH <p>8.1 ONLINE</p> <p>People can find the particular news and can get what they want</p> <p>8.2 OFFLINE</p> <p>People need to buy newspapers to read.</p>	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM <p>Before installing they feel it difficult to read news</p> <p>After using, they feel comfortable and user-friendly to get what they want.</p> <p>It conserves their time.</p>			

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	<ul style="list-style-type: none">• Registration through online• Registration through Gmail• Registration through mobile phone
FR-2	User installation	<ul style="list-style-type: none">• Application can download by play store• Or it can be viewed through website
FR-2	User Confirmation	<ul style="list-style-type: none">• Confirmation via Email• Confirmation via OTP
FR-3	User login	<ul style="list-style-type: none">• Login through email• Login through OTP• Login by username and password
FR-4	User information	<ul style="list-style-type: none">• It can pop up their preferred news article by setting the application settings

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none">• The option of filtering the content based on different category should be incorporated in the app to provide the audience wide taste and sensibilities. So, the users can read news that matters to them.• This will help the users to share news on various platforms such as Twitter and Facebook. This will not only give an amazing user experience and also will also increase the views.• When a user is not online due to some reason, he/she should have to access to the internet. Whenever the user is online the news content is downloaded in the cache memory of the app, this is how a user can access to the content offline.
NFR-2	Security	<ul style="list-style-type: none">• Authorised person can only manage database which contains the news article• The information about the users is secured safely.• People cannot watch news who are not registered with the application.
NFR-3	Reliability	<ul style="list-style-type: none">• This application publishes only the reliable content only with the help of API• How much time it takes to solve an issue

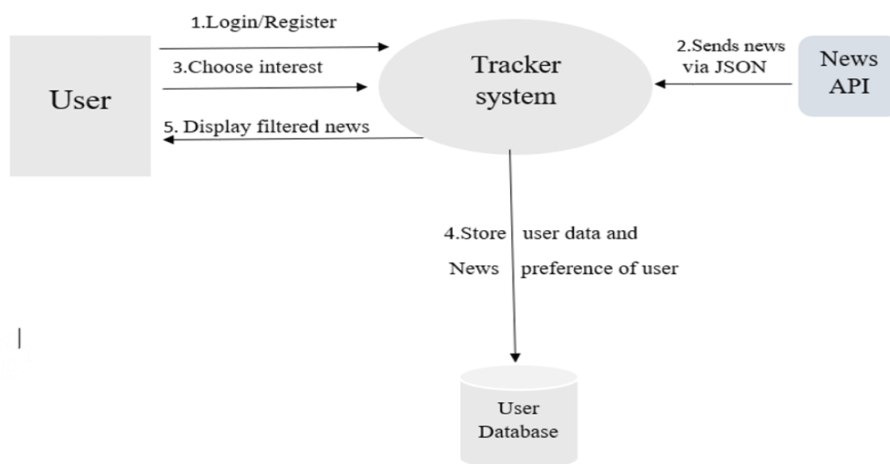
NFR-4	Performance	<ul style="list-style-type: none"> • Application performance refers to how well an app runs on a mobile device under various loads and circumstances. • It might require measuring how long an app takes to load, mitigating crashes during peaks in user activity, or monitoring battery usage. • App performance is an essential component of app development. • Potential users won't just uninstall an app when annoyed; they will likely seek a competitor to address their needs.
-------	--------------------	---

5. PROJECT DESIGN

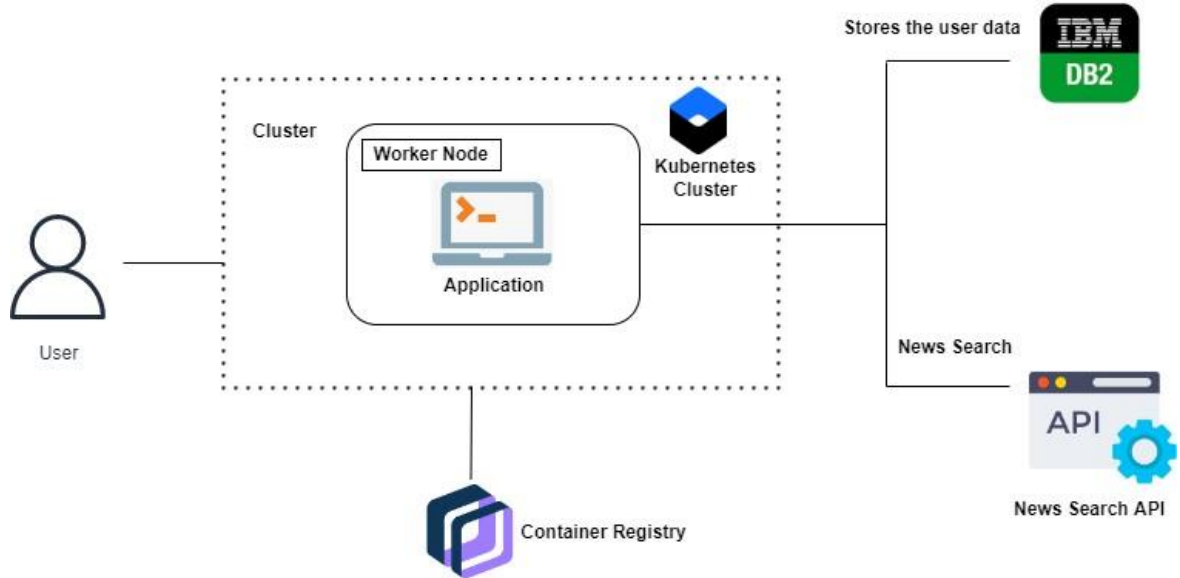
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clean DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system. What changes the information and where data is stored.

DATA FLOW DIAGRAM – NEWS TRACKER APPLICATION



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer (Mobile/web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password and confirm my password.	I can access my account/ dashboard	High	Sprint-1

		USN-2	As a user, I will receive confirmation email once I have registered for the application through Gmail.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail.		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password.		High	Sprint-1
	Dashboard	USN-5	As a user, I can enter the interests & choices of news I want to see for the first time in dashboard.		High	Sprint-2
		USN-6	As a user I can go through the feed of news filtered according to my wish.		High	Sprint-3
		USN-7	As a user, I can logout my account in settings.	I can click confirm to log out and end the session	Medium	Sprint-3

	Settings	USN-8	As a user, I can update my interests and choices in account settings.		Medium	Sprint-4
Customer Care Executive	Chat Bot / Query Section	USN-9	Solve issues brought up by client.		Medium	Sprint-4
Admin		USN10	Roll out updates and bug fixes.		High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey and Information Gathering	Literature survey on the selected project & gathering information by referring the technical papers, research publications, etc.	3 September 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user pains & gains, prepare the list of problem statements.	10 September 2022
Ideation Brain Storming	List by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility and importance.	16 September 2022

Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 September 2022
Problem Solution Fit	Prepare problem solution fit document.	28 September 2022
Solution Architecture	Prepare solution architecture document.	6 October 2022
Customer Journey	Prepare the customer journey maps, understand the user interactions and experiences with the application.	8 October 2022
Solution Requirement	Prepare the functional requirement document.	10 October 2022
Data Flow Diagram	Draw the data flow diagrams and submit for review.	15 October 2022
Technology Architecture	Prepare the technology architecture diagram.	20 October 2022
Prepare Milestone & Activity List	Prepare the milestones and activity list of the project.	28 October 2022
Project Development Delivery of Sprint-1,2,3 & 4	Develop and submit the developed code by testing it.	19 October 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Vignesh
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Vignesh
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	Ganesh
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Ganesh
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Vignesh
Sprint-2	Dashboard	USN-6	As a user I should be able to navigate And access all the features hassle free	2	High	Naveen
Sprint-2	Layout	USN-7	As a user I should be able to access the portal with different devices with the same comfort	2	High	Naveen
Sprint-3	Data Store and retrieval	USN-8	Get Data from API and store as JSON in DB2	3	High	Mohan Ram
Sprint-3		USN-9	Get bin data from API and store in DFS	2	High	Mohan Ram
Sprint-4	User Segregation and data access	USN-10	As a CC executive I should be able to uniquely identify the customer and offer help	1	Low	Ganesh
Sprint-4	Change code	USN-11	As a administrator I should be able to modify code according to the future requirements.	2	Medium	Vignesh
Sprint-4	Monitor the system	USN-12	As a administrator I should be able to monitor the cloud system and fix errors before customer.	1	High	Ganesh

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	4	19 Nov 2022

6.3 Reports from JIRA

The screenshot displays the Jira Software interface for a project named 'News Application'. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. A search bar is located on the right. The left sidebar shows the project's structure under 'PLANNING' and 'DEVELOPMENT', with 'Roadmap' selected. The main content area is titled 'Roadmap' and shows a grid view of the project's timeline. The grid has columns for months: NOV, DEC, and JAN '23. A vertical orange line indicates the current date. Under the 'NA-2 Web App UI' epic, there are four tasks: 'NA-3 Api integration', 'NA-4 Home screen', 'NA-5 Search screen', and 'NA-6 categories', all marked as 'TO DO'. A '+ Create Epic' button is visible below the tasks. At the bottom right, there are tabs for 'Today', 'Weeks', 'Months', and 'Quarters', and a 'Quickstart' button.

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

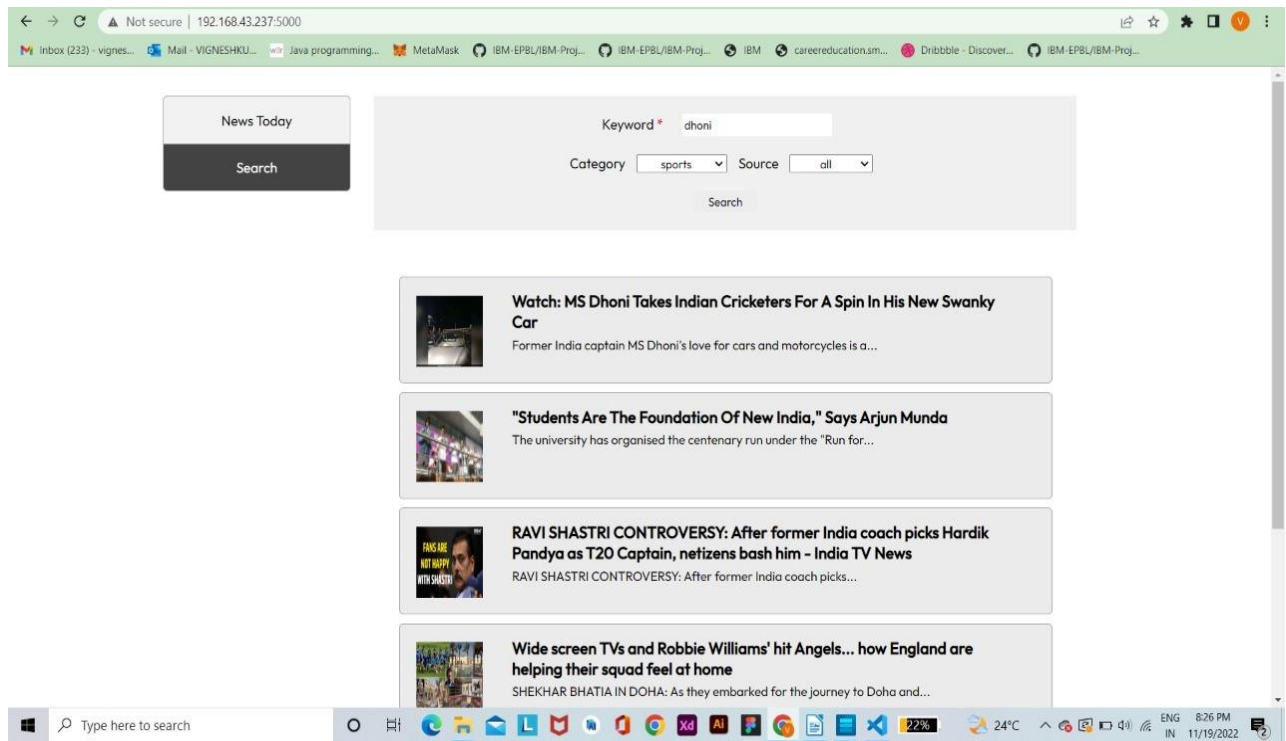
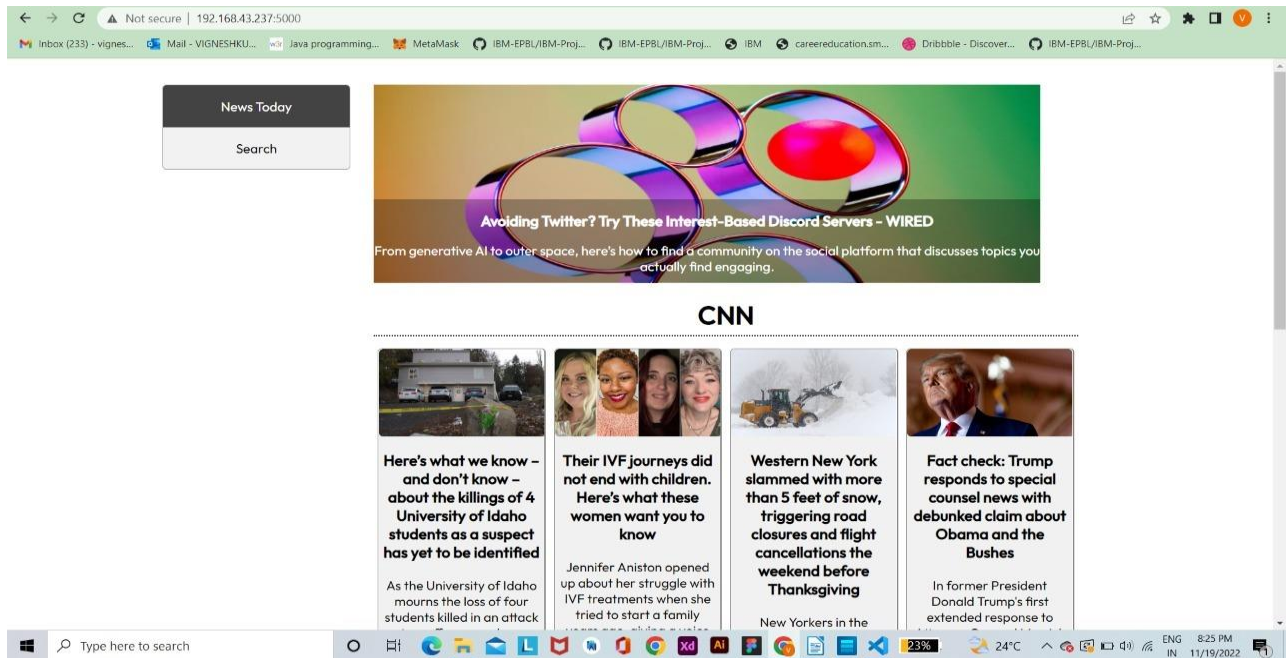
8. TESTING

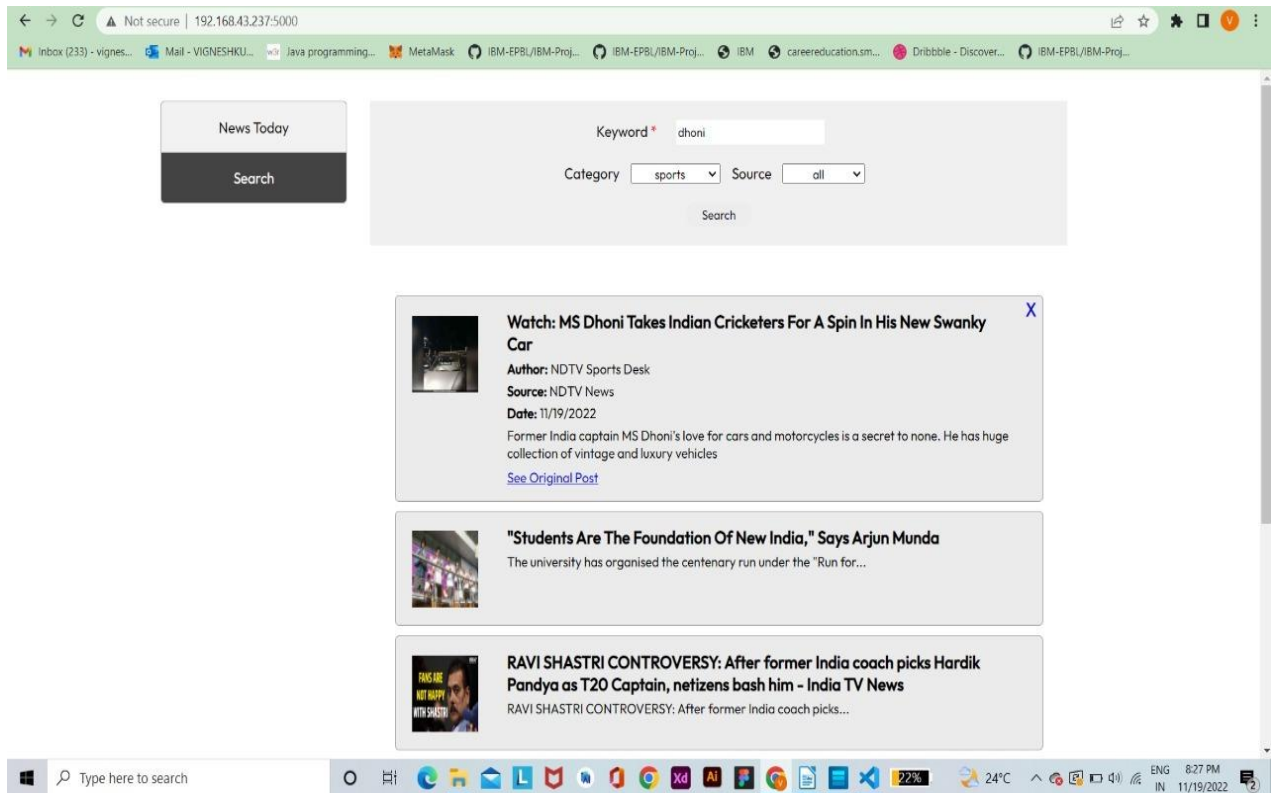
7.1 Test Cases

News Test Case:

```
def get_valid_articles(articles):
    required_keys = ["author", "description", "title", "url", "urlToImage",
                    "publishedAt"]
    # select only articles with the required keys not null
    valid_articles = [h for h in articles
                      if all(h[key] for key in required_keys)]
    valid_articles = [h for h in valid_articles
                      if h["source"]["name"] is not None]
    return valid_articles
```

8.1 User Acceptance Testing





9. RESULTS

News tracker application using cloud is developed and executed at the level of completed progress.

10. ADVANTAGES & DISADVANTAGES

Advantages:

- It provides higher accuracy through cross validation.
- High stability compare to Existing system.
- It is an user-friendly application.
- To explore and discover trending news and topics.
- Easily accessible and portable.
- Better user experience.
- Minute by minute updates of news.

Disadvantage:

- Occurrence of Advertisement disturb the user.
- Sometimes the news gives brief information.
- It works only through internet.
- Device fault may affect the application.
- Fake news may mislead the readers.

CONCLUSION:

We explored the feasibility of recognizing patterns of news reading interactions and evaluated three adaptive interface designs for different news reader types. We show that from their interaction log, a specific user can be recognized as one of three kinds. The reader types emerging from the online survey are well defined and distinct. The evaluation of the three variant interfaces suggests that different news reader types need different user interfaces. We have demonstrated a method for monitoring users' news reading behavior and inferring news reader type from it. In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalization of news apps.

FUTURE SCOPE:

In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalization of news apps.

13. APPENDIX

Source Code

main.py

```
from flask import Flask, jsonify, request
```

```
from newsapi import NewsApiClient
```

```
import re
```

```
from collections import Counter
```

```
from newsapi.newsapi_exception import NewsAPIException
```

```
app = Flask(__name__)
```

```
newsapi = NewsApiClient(api_key='68b80b58e40541468f4cf582b26463dd')
```

```
@app.route('/')  
def index():
```

```
    return app.send_static_file("index.html")
```

```
@app.route('/generic/')  
def get_generic_headlines():
```

```
    generic_headlines = newsapi.get_top_headlines(language='en')
```

```
    generic_headlines = generic_headlines['articles']
```

```
    generic_headlines = get_valid_articles(generic_headlines)[0:5]
```

```
    return jsonify(articles=generic_headlines)
```

```
@app.route('/cnn-fox/')  
def get_cnn_fox_headlines():
```

```
    cnn_headlines = newsapi.get_top_headlines(sources='cnn', language='en')
```

```
    fox_headlines = newsapi.get_top_headlines(sources='fox-news', language='en')
```

```
    cnn_headlines = cnn_headlines['articles']
```

```
    fox_headlines = fox_headlines['articles']
```

```
    cnn_headlines = get_valid_articles(cnn_headlines)[0:4]
```

```
    fox_headlines = get_valid_articles(fox_headlines)[0:4]
```

```
    hl = cnn_headlines + fox_headlines
```

```
    return jsonify(articles=hl)
```

```
@app.route('/word-cloud/')  
def get_word_cloud_words():
```

```
top_headlines = newsapi.get_top_headlines(language='en', page_size=100)
```

```
# extract titles
```

```
titles = [article["title"] for article in top_headlines["articles"]]
```

```
# extract words from titles
```

```
title_words = [re.findall(r'[a-zA-Z]+', t) for t in titles]
```

```
words = []
```

```
for title in title_words:
```

```
    words.extend(title)
```

```
# remove stop words
```

```
stop_words = []
```

```
with open("stopwords_en.txt", 'r') as file:
```

```
    for line in file:
```

```
        stop_words.append(line.rstrip())
```

```
words = [word for word in words if word.lower() not in stop_words]
```

```
cnt = Counter()
```

```
for word in words:
```

```
    cnt[word] += 1
```

```
word_cloud_words = [{"word": count[0], "size": count[1]*3}
```

```
                    for count in cnt.most_common(30)]
```

```
return jsonify(words=word_cloud_words)
```

```
@app.route('/get-sources/<category>')
```

```
def get_sources_for_category(category):
```

```
    sources = newsapi.get_sources(language='en')
```

```
    if category != "all":
```

```
        sources = newsapi.get_sources(category=category,
```

```
                                     language='en',
```

```
                                     country='us')
```



```
source_names = [s["name"] for s in sources["sources"]]
source_ids = [s["id"] for s in sources["sources"]]
source_names = source_names[0:10] if len(source_names) > 10 else source_names
source_ids = source_ids[0:10] if len(source_ids) > 10 else source_ids
return jsonify(source_names=source_names, source_ids=source_ids)
```

```
@app.route('/search/')
def get_search_results():
    args = request.args.to_dict()
    try:
        if args["src"] == "all":
            response = newsapi.get_everything(q=args["kw"],
                                              from_param=args["from"],
                                              to=args["to"],
                                              language="en",
                                              sort_by="publishedAt",
                                              page_size=100)
        else:
            response = newsapi.get_everything(q=args["kw"],
                                              sources=args["src"],
                                              from_param=args["from"],
                                              to=args["to"],
                                              language="en",
                                              sort_by="publishedAt",
                                              page_size=100)
    except NewsAPIException as ne:
        message = eval(str(ne))
        return jsonify(message)

# if success, then return first 15 articles
if response["status"] == "ok":
```

```
    response["articles"] = get_valid_articles(response["articles"])[0:15]
return jsonify(response)
```

```
def get_valid_articles(articles):
    required_keys = ["author", "description", "title", "url", "urlToImage",
                    "publishedAt"]
    # select only articles with the required keys not null
    valid_articles = [h for h in articles
                      if all(h[key] for key in required_keys)]
    valid_articles = [h for h in valid_articles
                      if h["source"]["name"] is not None]
    return valid_articles
```

```
if __name__ == '__main__':
    app.run(port=5000,host="0.0.0.0")
```

load.js

```
function load_everything() {
    // display only news section when first loaded
    document.getElementById("search").style.display = 'none';

    // define news/search tabs onclick behavior
    let tabs = document.getElementsByClassName("tab");
    for (let tab of tabs) {
        tab.onclick = function () {
            show_news_or_search(tab.id);
        }
    }
}
```

```

}

// load headlines slideshows, word cloud, CNN & Fox News
load_generic_headlines();
load_word_cloud_words();
load_cnn_fox_headlines();

// set default dates of the date selectors
set_default_date();

// load initial sources when category=all
load_sources("all");

// define category drop-down menu onclick behavior
let category = document.getElementById("category");
category.onchange = function () {
    load_sources(category.options[category.selectedIndex].value);
};

// define submit onclick behavior
document.getElementById("search-button").onclick = function (event) {
    event.preventDefault(); // prevent the page from refreshing

    // only submit the form when all fields have valid inputs
    if (document.getElementById("search-form").reportValidity()) {
        load_search_results();
    }
}

```

```
};
```

```
// define reset onclick behavior
```

```
document.getElementById("clear-button").onclick = reset_form;
```

```
}
```

```
function load_generic_headlines() {
```

```
  fetch("/generic/")
```

```
    .then(
```

```
      function(response) {
```

```
        if (response.status !== 200) {
```

```
          console.log(response);
```

```
          return;
```

```
        }
```

```
        response.json()
```

```
          .then(data => {
```

```
            display_generic_headlines(data.articles);
```

```
          });
```

```
        }
```

```
      )
```

```
    .catch(err => {
```

```
      console.log(err);
```

```
    })
```

```
}
```

```
function load_cnn_fox_headlines() {
```

```
fetch("/cnn-fox/")

.then(

  function (response) {

    if (response.status !== 200) {

      console.log(response);

      return;

    }

    response.json()

    .then(data => {

      display_cnn_fox_headlines(data.articles);

    });

  }

)

.catch(err => {

  console.log(err);

});

}
```

```
function load_word_cloud_words() {

  fetch("/word-cloud/")

  .then(

    function (response) {

      if (response.status !== 200) {

        console.log(response);

        return;
```

```
    }

    response.json()
      .then(data => {
        display_word_cloud(data.words);
      });
  }
)
.catch(err => {
  console.log(err);
})
}
```

```
function load_sources(value) {
  fetch(`/get-sources/${value}`)
    .then(
      function (response) {
        if (response.status !== 200) {
          console.log(response);
          return;
        }

        response.json()
          .then(data => {
            display_sources(data);
          });
      }
    );
}
```

```

        }
    )
    .catch(err => {
        console.log(err);
    })
}

```

```
function load_search_results() {
```

```

    let keyword = document.getElementById("keyword").value,
        from_date = document.getElementById("from-date").value,
        to_date = document.getElementById("to-date").value,
        category = document.getElementById("category").value,
        source = document.getElementById("source").value;

```

```
// check date range is valid
```

```
let start_date = new Date(from_date), finish_date = new Date(to_date);
```

```
if (finish_date < start_date) {
```

```
    alert("Incorrect time.");
```

```
    return;
```

```
}
```

```
fetch(`/search/?kw=${keyword}&from=${from_date}&to=${to_date}&cat=${category}&src=${source}`)
```

```
.then(
```

```
    function(response) {
```

```
        if (response.status !== 200) {
```

```
            console.log(response);
```

```

        return;
    }

    response.json()
        .then(data => {
            display_search_results(data);
        });
    }
)
.catch(err => {
    console.log(err);
})
}

```

display_headlines.js

```

function display_generic_headlines(jsonObj) {
    let slides = document.getElementsByClassName("generic-headline");
    let text_title = document.getElementsByClassName("gh-title");
    let text_desc = document.getElementsByClassName("gh-desc");
    for (let i = 0; i < slides.length; i++) { // hide everything at the beginning
        slides[i].style.display = "none";
    }

    slide_index ++;

    // reset slide_index (start the slideshow over)

    if (slide_index === slides.length) { slide_index = 1 }
}

```



```
// link entire div to news

let slide = document.getElementById("top-col-news").firstElementChild;
slide.href = jsonObj[slide_index-1].url;


// set image, title, description
slides[slide_index-1].getElementsByTagName("img")[0].src =
    jsonObj[slide_index-1].urlToImage;
text_title[slide_index-1].textContent = jsonObj[slide_index-1].title;
text_desc[slide_index-1].textContent = jsonObj[slide_index-1].description;
slides[slide_index-1].style.display = "block";
setTimeout(function() {display_generic_headlines(jsonObj)}, 4000);
}


function display_cnn_fox_headlines(jsonObj) {
    let hl_div = document.getElementsByClassName("hl-link");
    let hl_articles = document.getElementsByClassName("hl-text-container");

    for (let i=0; i<hl_articles.length; i++) {
        // link entire div to the news
        hl_div[i].href = jsonObj[i].url;

        // populate card with image, title, description
        let hl_article = hl_articles[i];
        hl_article.getElementsByTagName("img")[0].src = jsonObj[i].urlToImage;
        hl_article.getElementsByClassName("hl-title")[0].textContent = jsonObj[i].title;
        hl_article.getElementsByTagName("p")[0].textContent = jsonObj[i].description;
    }
}
```

```
}
```

```
function display_word_cloud(jsonObj){  
    // set the dimensions and margins of the graph  
    let width = 320, height = 250;  
  
    // append the svg object to the body of the page  
    let svg = d3.select("#word_cloud_container")  
        .append("svg")  
        .attr("width", width)  
        .attr("height", height);  
  
    // Constructs a new cloud layout instance.  
    // It run an algorithm to find the position of words that suits your requirements  
    let layout = d3.layout.cloud()  
        .size([width, height])  
        .words(jsonObj.map(function(d) { return {text: d.word, size:d.size}; })))  
        .padding(10) //space between words  
        .rotate(function() { return ~~(Math.random() * 2) * 90; })  
        .fontSize(function(d) { return d.size; }) // font size of words  
        .on("end", draw);  
    layout.start();  
  
    // This function takes the output of 'layout' above and draw the words  
    function draw(words) {  
        svg.append("g")  
            .attr("transform",
```

```

        "translate(" + layout.size()[0] / 2 + "," + layout.size()[1] / 2 + ")")
.selectAll("text")
.data(words)
.enter()
.append("text")
.style("font-size", function(d) { return d.size + "px";})
.style("fill", "#000000")
.attr("text-anchor", "middle")
.style("font-family", "Impact")
.attr("transform", function(d) {
    return "translate(" + [d.x, d.y] + ")rotate(" + d.rotate + ")";
})
.text(function(d) { return d.text; });
}
}

```

```

function display_sources(jsonObj) {
    let source_drop_down = document.getElementById("source");
    // remove all options
    for (let i = source_drop_down.length - 1; i > 0; i--) {
        source_drop_down.remove(i);
    }
    // add new sources for the category
    for (let i = 0; i < jsonObj.source_names.length; i++) {
        //for (let source of jsonObj.sources_names){
            let option = document.createElement("option");
            option.text = jsonObj.source_names[i];

```

```
    option.value = jsonObj.source_ids[i];  
    source_drop_down.add(option);  
  }  
}
```

search.js

```
function clear_search_results(show_less_div, show_more_div, results_div) {  
  // clear last-time search results  
  while (show_less_div.firstChild) {  
    show_less_div.firstChild.remove();  
  }  
  while (show_more_div.firstChild) {  
    show_more_div.firstChild.remove();  
  }  
  if (results_div.childElementCount > 2) {  
    results_div.removeChild(results_div.children[2]);  
  }  
}
```

```
function display_search_results(jsonObj) {  
  let results_div = document.getElementById("search-results"),  
      show_less_div = results_div.children[0],  
      show_more_div = results_div.children[1];  
  clear_search_results(show_less_div, show_more_div, results_div);  
  
  // hide articles in show-more by default  
  show_more_div.style.display = "none";
```

```
// check error

if (jsonObj.status === "error") {

    alert(jsonObj.message);

    return;

}
```

```
let articles = jsonObj.articles;
```

// if no article returned

```
if (articles.length === 0) {

    let no_results = document.createElement("p");

    no_results.classList.add("no-results");

    no_results.appendChild(document.createTextNode("No results"));

    results_div.appendChild(no_results);

    return;

}
```

// create card display for each article

```
for (let i = 0; i < articles.length; i++) {

    let result_card = create_result_card(articles[i]);

    let result_card_expanded = create_result_card_expanded(articles[i]);

    // define result card onclick behavior

    result_card.onclick = function () {

        result_card.style.display = "none";

        result_card_expanded.style.display = "grid";

    };

}
```

```
// add close button

let close_sign = document.createElement("a");

close_sign.href = "#";

close_sign.classList.add("close-sign");

close_sign.textContent = "x";

// define close button onclick behavior

close_sign.onclick = function () {

    result_card.style.display = "grid";

    result_card_expanded.style.display = "none";

};

result_card_expanded.appendChild(close_sign);


// put first 5 articles in show-less div, and the rest in show-more div

if (i < 5) {

    show_less_div.appendChild(result_card);

    show_less_div.appendChild(result_card_expanded);

}

else {

    show_more_div.appendChild(result_card);

    show_more_div.appendChild(result_card_expanded);

}

}

if (articles.length > 5) {

    // implement show-more show-less

    let show_more_less = document.createElement("button");

    show_more_less.id = "show-more-less-button";
```

```
show_more_less.classList.add("show-more");

show_more_less.textContent = "Show More";

results_div.appendChild(show_more_less);


show_more_less.onclick = function () {
    if (show_more_less.className === "show-more") {
        show_more_div.style.display = "block";
        show_more_less.classList.remove("show-more");
        show_more_less.classList.add("show-less");
        show_more_less.textContent = "Show Less";
    } else {
        show_more_div.style.display = "none";
        show_more_less.classList.remove("show-less");
        show_more_less.classList.add("show-more");
        show_more_less.textContent = "Show More";
    }
}
}
```

create.js

```
function create_result_card_expanded(article) {
    let result_card = create_result_card(article);
    result_card.classList.add("expanded");
    result_card.style.display = "none";

    /* change to detailed information */
}
```

```
let text_div = result_card.getElementsByClassName("result-text-div")[0];

// remove one line description
text_div.removeChild(text_div.childNodes[1]);

// add author info
let author_p = document.createElement("p");
author_p.innerHTML = `Author: ${article.author}`;
text_div.appendChild(author_p);

// add source info
let source_p = document.createElement("p");
source_p.innerHTML = `Source: ${article.source.name}`;
text_div.appendChild(source_p);

// add date info
let date_p = document.createElement("p");
let date = new Date(article.publishedAt);

date = `${("0" + (date.getMonth() + 1)).slice(-2)}/${("0" + date.getDate()).slice(-2)}/${date.getFullYear()}`;

date_p.innerHTML = `Date: ${date}`;
text_div.appendChild(date_p);

// add detailed description
let desc_p = document.createElement("p");
desc_p.classList.add("desc-p");
desc_p.textContent = article.description;
text_div.appendChild(desc_p);
```



```
// add link to original news post

let link_to_news = document.createElement("a");

link_to_news.href = article.url;

link_to_news.target = "_blank";

link_to_news.textContent = "See Original Post";

text_div.appendChild(link_to_news);


return result_card;
}


function create_result_card(article) {

    let result_card = document.createElement("div");

    result_card.classList.add("result-card");


    // create img div for result card

    let result_img_div = document.createElement("div");

    result_img_div.classList.add("result-img-div");


    let result_img = document.createElement("img");

    result_img.src = article.urlToImage;


    result_img_div.appendChild(result_img);

    result_card.appendChild(result_img_div);


    // create text div for result card

    let result_text_div = document.createElement("div");

    result_text_div.classList.add("result-text-div");
```

```
let result_title = document.createElement("h3");
result_title.textContent = article.title;
let result_desc = document.createElement("p");
let desc = article.description.slice(0, 70);
// remove html tags from returned description
desc = desc.replace(/(<\w+>)+/, "");
// display only one line with ellipsis cut off
desc = desc.replace(/^(W*\w+\.W*$/, "...");
result_desc.textContent = desc;

result_text_div.appendChild(result_title);
result_text_div.appendChild(result_desc);

// put img and text in result cards
result_card.appendChild(result_img_div);
result_card.appendChild(result_text_div);

return result_card;
}

function show_news_or_search(clicked_tab_id) {
  let news_div = document.getElementById("news");
  let search_div = document.getElementById("search");

  let tab_news = document.getElementById("tab-news");
  let tab_search = document.getElementById("tab-search");
```

```
if (tab_news.classList.contains("active") && clicked_tab_id === "tab-search") {  
    tab_news.classList.remove("active");  
    tab_search.classList.add("active");  
    search_div.style.display = "flex";  
    news_div.style.display = "none";  
}  
  
else if (tab_search.classList.contains("active") && clicked_tab_id === "tab-news") {  
    tab_news.classList.add("active");  
    tab_search.classList.remove("active");  
    news_div.style.display = "flex";  
    search_div.style.display = "none";  
}  
}  
  
function set_default_date() {  
    let curr_day = new Date();  
    let week_ago = new Date();  
    week_ago.setDate(curr_day.getDate() - 7);  
    document.getElementById("from-date").valueAsDate = week_ago;  
    document.getElementById("to-date").valueAsDate = curr_day;  
}  
  
function reset_form() {  
    // reset the form  
  
    document.getElementById("search-form").reset();  
  
    // set default dates
```

```
set_default_date();

// clears previous search results

let results_div = document.getElementById("search-results"),

    show_less_div = results_div.children[0],

    show_more_div = results_div.children[1];

clear_search_results(show_less_div, show_more_div, results_div);

}
```

```
window.onload = load_everything;
```

```
var slide_index = 0;
```

index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Top News</title>

    <link rel="stylesheet" type="text/css" href="static/css/style.css">

    <link rel="shortcut icon" href="#" />

</head>

<body>

<div id="body-div">

    <div id="tabs">

        <button class="tab active" id="tab-news" style="font-family: 'Outfit', sans-serif;">

            News Today
```

```
</button>
```

```
<button class="tab" id="tab-search" style="font-family: 'Outfit', sans-serif;">
```

```
    Search
```

```
</button>
```

```
</div>
```

```
<div id="news">
```

```
    <div id="top-row">
```

```
        <div class="top-col" id="top-col-news">
```

```
            <a target="_blank">
```

```
                <div id="slideshow-container" class="top-card" style="width:800px">
```

```
                    <div class="generic-headline">
```

```
                        <img src="" class="gh-img">
```

```
                        <div class="text-block">
```

```
                            <h2 class="gh-title"></h2>
```

```
                            <p class="gh-desc"></p>
```

```
                        </div>
```

```
                    </div>
```

```
                <div class="generic-headline">
```

```
                    <img src="" class="gh-img">
```

```
                    <div class="text-block">
```

```
                        <h2 class="gh-title"></h2>
```

```
                        <p class="gh-desc"></p>
```

```
                    </div>
```

```
                </div>
```

```
            <div class="generic-headline">
```

```
                <img src="" class="gh-img">
```

```
                <div class="text-block">
```

```
        <h2 class="gh-title"></h2>

        <p class="gh-desc"></p>

    </div>

</div>

<div class="generic-headline">

    <img src="" class="gh-img">

    <div class="text-block">

        <h2 class="gh-title"></h2>

        <p class="gh-desc"></p>

    </div>

</div>

<div class="generic-headline">

    <img src="" class="gh-img">

    <div class="text-block">

        <h2 class="gh-title"></h2>

        <p class="gh-desc"></p>

    </div>

</div>

</div>

</a>

</div>

<!-- <div class="top-col" id="top-col-wc">

    <div id="word_cloud_container" class="top-card">

        </div>

    </div> -->

</div>

<div id="cnn-fox">
```

```
<h1 class="cnn-fox-header">CNN</h1>

<div id="cnn-news" class="headlines">

  <div class="hl-col">

    <a class="hl-link" target="_blank">

      <div class="hl-card">

        <div class="hl-text-container">

          <img src="">

          <h3 class="hl-title"></h3>

          <p></p>

          <button class="read">Read more</button>

        </div>

      </div>

    </a>

  </div>

  <div class="hl-col">

    <a class="hl-link" target="_blank">

      <div class="hl-card">

        <div class="hl-text-container">

          <img src="">

          <h3 class="hl-title"></h3>

          <p></p>

          <button class="read">Read more</button>

        </div>

      </div>

    </a>

  </div>

  <div class="hl-col">
```

```
<a class="hl-link" target="_blank">

  <div class="hl-card">

    <div class="hl-text-container">

      <img src="">

      <h3 class="hl-title"></h3>

      <p></p>

      <button class="read">Read more</button>

    </div>

  </div>

</a>

</div>

<div class="hl-col">

  <a class="hl-link" target="_blank">

    <div class="hl-card">

      <div class="hl-text-container">

        <img src="">

        <h3 class="hl-title"></h3>

        <p></p>

        <button class="read">Read more</button>

      </div>

    </div>

  </a>

</div>

</div>

<h1 class="cnn-fox-header">Fox News</h1>

<div id="fox-news" class="headlines">

  <div class="hl-col">
```



```
<a class="hl-link" target="_blank">

  <div class="hl-card">

    <div class='hl-text-container'>

      <img src="">

      <h3 class='hl-title'></h3>

      <p></p>

      <button class="read">Read more</button>

    </div>

  </div>

</a>

</div>

<div class="hl-col">

  <a class="hl-link" target="_blank">

    <div class="hl-card">

      <div class='hl-text-container'>

        <img src="">

        <h3 class='hl-title'></h3>

        <p></p>

        <button class="read">Read more</button>

      </div>

    </div>

  </a>

</div>

<div class="hl-col">

  <a class="hl-link" target="_blank">

    <div class="hl-card">

      <div class='hl-text-container'>
```

```
        <img src="">

        <h3 class='hl-title'></h3>

        <p></p>

        <button class="read">Read more</button>

    </div>

</div>

</a>

</div>

<div class="hl-col">

    <a class="hl-link" target="_blank">

        <div class="hl-card">

            <div class='hl-text-container'>

                <img src="">

                <h3 class='hl-title'></h3>

                <p></p>

                <button class="read">Read more</button>

            </div>

        </div>

    </a>

</div>

</div>

<div id="search">

    <div id="search-box">

        <form id="search-form" onsubmit="load_search_results()">

            <div id="search-kw-date">
```

```
<label for="keyword">Keyword</label>

  <input type="text" id="keyword" name="keyword" required>

<div style="display: none">

<label for="from-date">From</label>

  <input type="date" id="from-date" name="from-date" required>

<label for="to-date">To</label>

  <input type="date" id="to-date" name="to-date" required>

</div>

</div>

<br>

<div id="search-cat-src">

  <label for="category">Category</label>

  <select id="category" name="category">

    <option value="all">all</option>

    <option value="business">business</option>

    <option value="entertainment">entertainment</option>

    <option value="general">general</option>

    <option value="health">health</option>

    <option value="science">science</option>

    <option value="sports">sports</option>

    <option value="technology">technology</option>

  </select>

  <label for="source">Source</label>

  <select id="source">

    <option value="all" id="all">all</option>

  </select>

</div>
```

```
<br>

<div id="search-submit">

    <input type="submit" id="search-button" value="Search">

</div>

</form>

</div>

<div id="search-results">

    <div id="show-less-div"></div>

    <div id="show-more-div"></div>

</div>

</div>

</div>

</body>

<script type="application/javascript" src="static/js/index.js"></script>

<!-- Load d3.js -->

<script src="https://d3js.org/d3.v4.js"></script>

<!-- Load d3-cloud -->

<script src="https://cdn.jsdelivr.net/gh/holtzy/D3-graph-gallery@master/LIB/d3.layout.cloud.js"></script>

</html>
```

style.css

```
@import
url('https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500&display=swap');

* {

    box-sizing: border-box;

    font-family: 'Outfit', sans-serif;
```

```
}  
  
#body-div {  
    display: flex;  
    width: 75%;  
    margin: auto auto auto 10%;  
}  
  
#tabs {  
    margin-top: 25px;  
    text-align: center;  
    flex: 1;  
}  
  
.tab {  
    font-family: Georgia, serif;  
    width: 80%;  
    background-color: #f2f2f2;  
    padding: 16px 20px;  
    text-align: center;  
    font-size: 16px;  
    cursor: pointer;  
    border: #aaaaaa solid 1px;  
    transition: .3s;  
}  
  
.tab.active {  
    background-color: #434343;  
    color: white;  
}
```

```
.tab:hover {  
    background-color: #007bff;  
    color: white;  
    cursor: auto;  
}  
  
.read{  
    background-color: #007bff;  
    color: white;  
    border: none;  
    height: 40px;  
    width: 80px;  
    border-radius: 10px;  
}  
  
#tab-news {  
    border-radius: 5px 5px 0 0;  
}  
  
#tab-search {  
    margin-top: -1px;  
    border-radius: 0 0 5px 5px;  
}  
  
#news {  
    display: flex;  
    flex: 3;  
    margin-bottom: 5%;  
    flex-direction: column;  
}
```

```
.top-col {  
    float: left;  
    margin-bottom: 0;  
}  
  
#top-col-news {  
    width: 55%;  
    margin-right: 10px;  
}  
  
#top-col-wc {  
    width: 35%;  
    display: block;  
    margin-left: 10px;  
}  
  
svg {  
    background-color: #f2f2f2;  
}  
  
.generic-headline {  
    position: relative;  
}  
  
.generic-headline .gh-img {  
    width: 100%;  
    height: 250px;  
    bottom: 0;  
    margin-bottom: -5px;  
}  
  
.text-block {
```

```
position: absolute;
background-color: rgba(0, 0, 0, 0.3);
color: white;
bottom: 0;
margin-bottom: 0;
width: 100%;
}
.text-block p {
    font-size: 16px;
    margin-bottom: 10px;
}
.text-block h2 {
    font-size: 18px;
}
#top-row {
    margin: 25px 0 0 0;
}
.top-card {
    text-align: center;
}
#top-content {
    margin-bottom: 50px;
}
.cnn-fox-header {
    margin-bottom: 0;
    text-align: center;
```



```
}  
  
.headlines {  
    display: flex;  
    border-top: 2px dotted;  
    margin-top: 5px;  
    padding-top: 10px;  
}  
  
.hl-col {  
    flex: 1;  
}  
  
a.hl-link:any-link {  
    text-decoration: none;  
}  
  
.hl-card {  
    margin: 5px;  
    border-style: solid;  
    border-width: 1px 2px 2px 1px;  
    border-color: #828282;  
    border-radius: 5px;  
}  
  
.hl-card img {  
    width: 101%;  
    height: 22%;  
    border-radius: 5px;  
}  
  
.hl-text-container {
```

```
text-align: center;

/* TODO: use flexbox to auto-adjust height */

height: 500px;

overflow: hidden;

color: black;

background-color: #f1f1f1;

border-radius: 5px;
}

.hl-text-container .hl-title, .hl-text-container p {

padding: 5px;

margin: 10px 0 10px 0;
}

#search {

margin-top: 25px;

margin-bottom: 5%;

flex: 3;

flex-direction: column;
}

#search-box {

flex: 1;
}

#search-form {

background-color: #f0f0f0;

padding: 20px;

text-align: center;
}
```

```
#search-kw-date label:after {
```

```
    content: " *";
```

```
    color: red;
```

```
}
```

```
#search-kw-date input {
```

```
    border: none;
```

```
    margin: 0 20px 0 20px;
```

```
    height: 25px;
```

```
    border-radius: 1px;
```

```
}
```

```
#search-cat-src select {
```

```
    text-align: center;
```

```
    margin: 0 10px 0 10px;
```

```
}
```

```
#search-cat-src #source {
```

```
    width: 100px;
```

```
}
```

```
#search-submit input {
```

```
    background-color: #ededed;
```

```
    border: none;
```

```
    width: 80px;
```

```
    height: 30px;
```

```
    border-radius: 10px;
```

```
    height: 25px;
```

```
    margin: 0 1% 0 1%;
```

```
    font-size: 10pt;
```

```
    transition: .3s;
}

#search-submit input:hover {
    background-color: #007bff;
    color: white;
}

#search-results {
    position: relative;
    margin-top: 5%;
}

#search-results .no-results {
    text-align: center;
    width: 50%;
    margin: 0 auto;
}

#search-results .result-card {
    border: 1px solid #9f9b9b;
    border-radius: 5px;
    height: 120px;
    background-color: #ebebeb;
    margin: 10px 30px;
    display: grid;
    grid-template-columns: 120px auto 20px;
    z-index: 0;
}

#search-results .result-card:hover {
```

```
    box-shadow: 0 10px 25px 0 #787878;

    cursor: pointer;

    z-index: 1;
}

#search-results .result-card.expanded {

    height: 100%;

    overflow: auto;
}

.result-img-div {

    float: left;

    width: 120px;

    height: 100%;
}

.result-img-div img {

    margin: 20px;

    width: 80px;

    height: 80px;
}

.result-text-div {

    padding: 15px;

    height: 100%;
}

.result-text-div h3 {

    margin: 0;

    font-size: 14pt;
}
```

```
.result-text-div p, .result-text-div a {  
    margin: 5px 0 5px 0;  
    font-size: 11pt;  
}
```

```
.result-text-div p.desc-p{  
    display: -webkit-box;  
    overflow : hidden;  
    text-overflow: ellipsis;  
    -webkit-line-clamp: 3;  
    -webkit-box-orient: vertical;  
}
```

```
.close-sign {  
    font-size: 20pt;  
    font-family: Arial, sans-serif;  
    text-decoration: none;  
    margin-top: -3px;  
}
```

```
#show-more-less-button {  
    display: block;  
    margin: 0 auto;  
    width: 13%;  
    text-align: center;  
    background-color: #e6e6e6;  
    border: none;  
    padding: 10px;  
}
```

```
#show-more-less-button:hover {  
  background-color: #3c963c;  
  color: white;  
}
```

app.yaml

```
runtime: python37
```

GitHub & Project Demo Link

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-25115-1659953875>

Project Demo Link:

- <https://news-tracker-application-1368100.s3.amazonaws.com/object-storage.appdomain.cloud/Demo-Link-News-Tracker-Application.mp4>