NEWS TRACKER APPLICATION

IBM PROJECT

TEAM_ID:PNT2022TMID12184

Submitted by

113319104032	**JAYANT H**
113319104005	**AKASH K**
113319104052	**MURALI DHARAN S**
113319104074	**SANJAY R**
113319104103	**VENKATASRINIVAS P.V**

in partial fulfillment for the award of

the degreeof

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

VELAMMAL INSTITUTE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

NOV 2022

Project Report

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
    1. Performance Metrics

1. INTRODUCTION

1.1 Project Overview

News Tracker is an application that primarily focuses on enhancing and optimizing the way of user interaction with the news stories. It is more user friendly in a way that a user can personalize his/her own interests in this application. The study process began with how other applications developed with respect to personalization and giving users a curated content. This step allowed the study analysis identify the disadvantages and make it as advantages in this application. The idea of personalization is that it keeps track of the user's interests and accordingly change the home page for the users to give them a personalized experience based on what they set as their preferences. The application also has an added advantage that any user using this application might not be able to feel discomfort as there are no advertisement which many applications failed to meet up the expectations of a user. The application stands out off the crowd in terms of personalization, better search and able to view in multiple languages.

1.2 Purpose

News tracker serves the purpose of delivering a better application to the users and enhanced user experience and user interface. The ultimate purpose is giving the user a personaslized application with an ability to better search and this application doesn't function on one language and thus enabling users to use this application worldwide. The users should be displayed with clean, personalized content on the home page with zero advertisements. The need for this project is to deliver news stories without any distractions and getting to see the content only the users like.

2. LITERATURE SURVEY

2.1 Existing Problem

Newspapers were the leading source and first physical source of news for people. Even though the habit of reading newspapers has decreased significantly, the information sector is still alive and well.

Several advancements in the technology sector first enabled smartphones to display latest and updated news on reader's demand. Personalisation of news access should extend beyond 'what' the user sees to 'how' they access it. Popular applications like **Google News**, **BBC** and **DailyHunt** provide personalised and curated content to users. Users select topics to follow and the system generates a news feed for them to read and also provides a summary for each article.

The personalisation news apps can be achieved by making the User Interface adaptable. An adaptable user interface will allow the system to adapt to the way the users read and provide summaries of articles.

Existing System:

Google News:

- Under each headline, the "Full Coverage" button lets users round up coverage from the best news sources on a given topic—something discerning readers value in an era of intense partisanship and concern about biased reporting.

- Ability to subscribe to specific news sources and read them directly from the app.

- Option to purchase premium subscriptions to news outlets.

- Interconnected updates on the latest developments for existing news updates.

2.2 References

- https://news.google.com
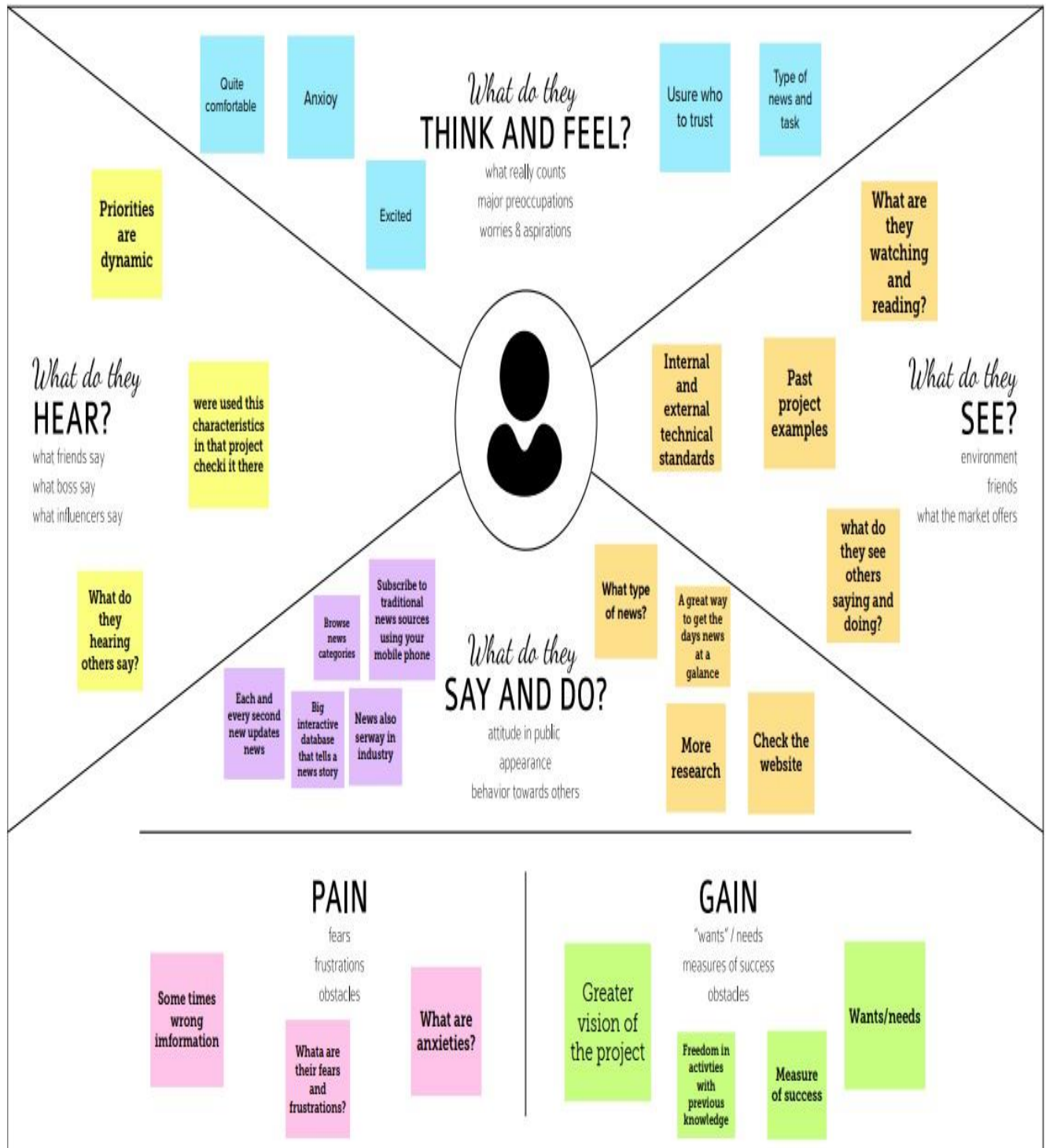- https://www.msn.com/
- http://news.yahoo.com/
- http://twitter.com/

2.3 Problem Statement Definition

News articles are accessed via smartphones and tablets more now than ever. This has created a need for providing personalised feed of news articles for users. A recent study by Reuters Institute for the Study of Journalism's Digital News Report said that about 82 percent of Indians read the news online, out of which 73 percent do it on their smartphones.

Curated news feed has been widely used as the tool for sourcing a variety of content from different sources on a report and providing a 360-degree review of the story. The demand for news curation has increased over the years and users looks for SMEs to help them find quality and informative content without having to spend much time browsing or scan multiple articles themselves.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation and Brainstorming

NEWS TRACKER APPLICATION

Design
Layout · Images · Colors · Fonts

Front End
User Interface · HTML,CSSJS · API · Responsible Design

Back End
Database · IBM WATSON · IBM CLoud · Docker

User Experience
Login · UNDERSTAND · USER NEEDS · search

ROLE AND RESPONSIBILITY

01
Akash k
Create and set up the necessary environments for the implementation of the application.

02
Jayant H
Development of application in IBM cloud database and connecting it to the Flask framework.

03
Murali Dharan S
Implementation of UI interaction with Front-end Development

04
Sanjay R
Project Designing phase 1 and project plannning phase.

05
Venkatasrinivas P V
Project Designing phase 2 and project development phase.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	With lives getting busier by the day, people find it difficult to keep up with current affairs. In some cases, users find it hard to trust the source of the information they read and hence customer base also goes down gradually. This can be overcome by providing the users curated or personalised content based on their interests and Subject Matter Experts can be brought in to determine if the sources are credible or not.

2.	Idea / Solution description	Several events occur globally. Users find it cumbersome to search for topics or specific articles. This system will allow users to select a few topics or domains and they will be provided a feed with articles and reports from those domains.
3.	Novelty / Uniqueness	The credibility of the sources mentioned in the feed cannot be determined without the verification of content by Journalists, Reporters and Subject Matter Experts. The system's feed will be void of falsified and misleading reports and articles.
4.	Social Impact / Customer Satisfaction	News readers have reduced because of increasing falsified information and misleading titles in reports. Giving them an accurate report of the event will increase customer base, resulting in an increased level of general knowledge amongst the general populous.
5.	Business Model (Revenue Model)	Website Monetisation by allowing Web Banners can increase the revenue generated. Providing a subscription based premium articles and removing advertisements similar to Harvard Business Review.
6.	Scalability of the Solution	The application is designed and constructed using a scalable architecture. Hence, the system will be able to meet any amount of expected traffic, working seamlessly across pages.

3.3 Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)　CS

Our customers are the one who needs to know the updated News world wide. Customers age is between 10 – 60 years, in which they are easily able to understand the overall layer of the application.

6. CUSTOMER CONSTRAINTS

The following constraints prevents our customer from taking action,
1. Slow or no internet connection.
2. Low hardware and software device.
3. Older version of customer's browser.
4. No software updates available to the customer.

5. AVAILABLE SOLUTIONS

1. Customer needs to check for updates to the latest version of their browser.
2. Customer needs to upgrade to their latest hardware device along with new software with updates.
3. Customer must change the network provider service for better network speed.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS　J&P

1. The user interface of the application can be built simple and neat so that the end user can easily understand the prototype.
2. There can be an email subscription so that we can kept connected to the customer.

9. PROBLEM ROOT CAUSE　RC

The real reason is that customer cannot carry a newspaper of television along with them, So the end user need not carry these thing when the user can see and subscribe to the news letter and feed.

7. BEHAVIOUR　BE

Our customer searches for a news feed website rather than downloading it into the customer's device, then they can able to track the day-to-day news

Focus on J&P, tap into BE, understand RC & Identify strong Tr

3. TRIGGERS　TR

The title and thumbnail provided by the news provider shows an interest on the followers that triggers the customer to click and view the news.

4. EMOTIONS: BEFORE / AFTER　EM

Customer feel unsatisfied when the application does not meet the requirements of the customer like navigating to the next page, reversing the actions and guiding throughout the browsing.

10. YOUR SOLUTION　SL

Our solution is to provide a clean interface to the customer to view and post messages to the following news feeds.

8. CHANNELS of BEHAVIOUR　CH

8.1 ONLINE
When customer device is online, they can able to download the news and subscribe to their respective news channels.

8.2 OFFLINE
When customer device is offline, they can able to open and read the news by scrolling the news feed.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR – 1	User Registration	Registration through Form Registration through Gmail
FR – 2	User Confirmation	Confirmation via Email Confirmation via OTP
FR – 3	Find Content based on Topic and Region	Users should be able to find articles or content based on the region they are in or using a search bar for content based and region-based searches
FR – 4	User – friendly Ads	Advertisements are great revenue generators and they should be displayed as web banners.
FR – 5	Real time news updates	An auto refresh feature should be enabled for users to read latest content without a lapse in time
FR – 6	Allowing subscriptions and registrations	Adding subscription feature and registrations can be used to track user's loyalty and the type of content they're interested in

4.2 Non-functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	● Simple and elegant User Interface ● The feed and content should be aligned well and application should be responsive to all devices
NFR-2	**Security**	● Strong password requirements should be set ● Two-step verification for every login ● No sensitive data should be stored
NFR-3	**Reliability**	● The content of the feed should be verified by reporters
NFR-4	**Performance**	● Seamless and smooth transition of pages ● Low lapse of time in case of lower end devices

NFR-5	**Availability**	- Web application delivered over the internet - Low downtime for production changes and maintenance
NFR-6	**Scalability**	Scalable and simple application that is made available to customers and users

5. PROJECT DESIGN

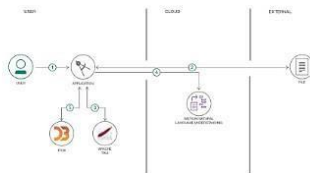## 5.1 Data Flow Diagrams

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear 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.

Example: Flow Diagram

Flow

1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

Example: DFD Level 0 (Industry Standard)
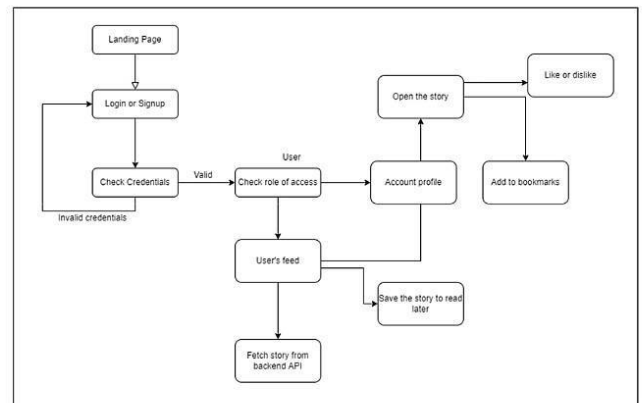
5.2 Solution & Technical Architecture

Solution Architecture: Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to: • Find the best tech solution to solve existing business problems.

- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements to the project.
- Provide specifications according to which the solution is defined, managed, and delivered to the customers.
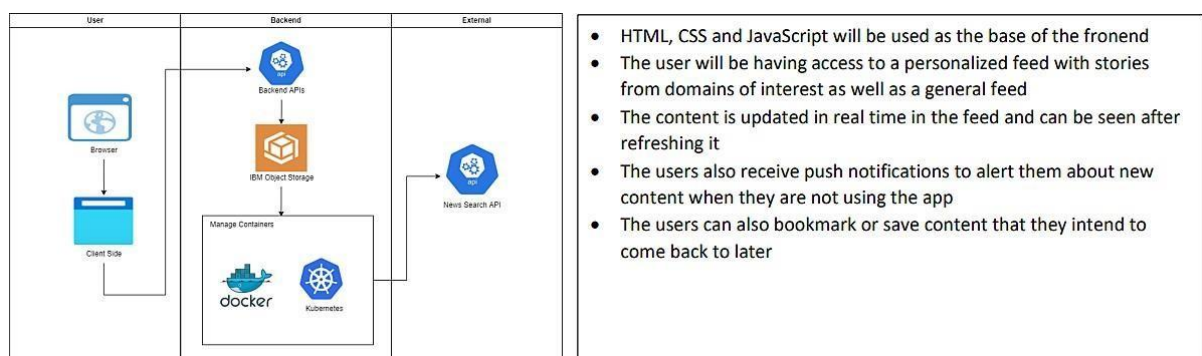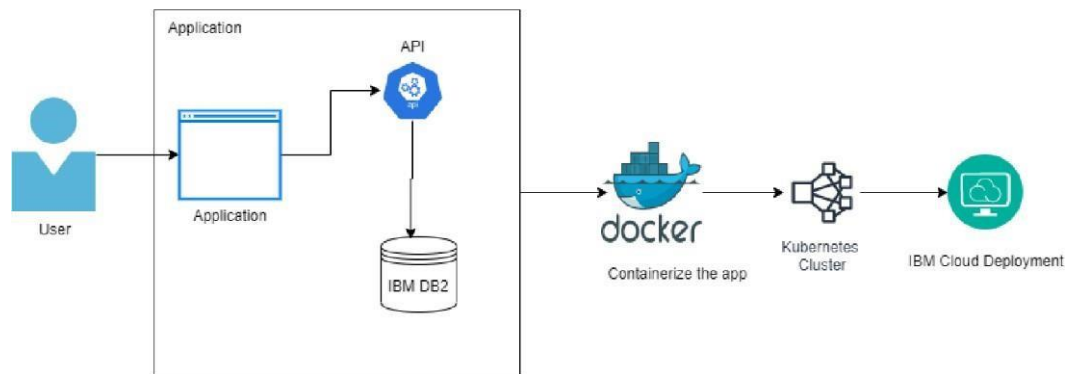
- HTML, CSS and JavaScript will be used as the base of the fronend
- The user will be having access to a personalized feed with stories from domains of interest as well as a general feed
- The content is updated in real time in the feed and can be seen after refreshing it
- The users also receive push notifications to alert them about new content when they are not using the app
- The users can also bookmark or save content that they intend to come back to later

Components and Technologies

S.No	Component	Description	Technology
1.	User Interface	The user should be able to access the application through a web browser in a range of devices like desktops, mobile phones, tablets etc	HTML, CSS, React Js.
2.	Application Logic	Logic for the backend processes in the application	Flask (Python)
3.	Database	Data Type, Configurations etc.	MySQL
4.	Cloud Database	Database Service on Cloud	IBM DB2
5.	File Storage	File storage requirements	IBM Block Storage

6.	External API	To fetch news articles that are updated in real time	News Search API
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Docker, Kubernetes.

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Python
2.	Security Implementations	IAM controls provided by the IBM cloud	Encryptions, IAM Controls
3.	Scalable Architecture	3 tier architecture	IBM container registry, IBM DB2 cloud
4.	Availability	Distributed cloud, and automatic load balancing with containerization	Docker, Kubernetes
5.	Performance	High performance as a result of load balancing and containerization	Container registry, Kubernetes

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming 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	I can receive confirmation email & click confirm	Medium	Sprint-1

		USN-3	As a user, I can register for the application through Google	I can register & access the dashboard with Google Login	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I'm redirected to dashboard if entered credentials are valid	High	Sprint-1
	Dashboard	USN-5	The updated feed with latest stories fetched from the API are displayed	I'm able to view latest stories in my feed	High	Sprint-1
Customer (Web user)	Search Bar	USN-6	Users searches for stories or topic in the search bar	I'm able to view relevant stories in my feed	Medium	Sprint-2
		USN-7	Users can apply filters to their searches such as date, publisher etc	I'm able to apply filters to my search results for more accurate results	Low	Sprint-1
Customer Care Executive	Chatbot	USN-8	A functional chatbot can respond to user queries in real time	I'm able to clear my queries with the help of the chatbot	Medium	Sprint-3
Administrator	Server	USN-9	Administrator verifies and validates the news available in the database	Fakes and invalid news will be rejected and removed from the database	High	Sprint-4
		USN-10	Administrator provides news articles with video and audio content	I am able to view the articles with video and audio content	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requireme nt (Epic)	User Story/Task User Story/No		User Story / Numbe r	Priority	Team Membe rs
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	Murali Dharan,Akash.Sanjay
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	Medium	Murali Dharan ,Sanjay
Sprint-2		USN-3	As a user, I can register for the application through Google	1	Low	Akash
Sprint-1	Login	USN-4	As a user, I can log into the applicatio n by entering email & password	3	High	Sanjay,Akash,Venkatasrini vas
Sprint-1	Dashboard	USN-5	The updated feed with latest stories fetched from the API are displayed	3	High	Sanjay,Jayant,Akash

| Sprint-2	Search Bar	USN-6	Users searches for stories or topic in the search bar	2	Medium	Akash,Venkatasrinivas
		USN-7	Users can apply filters to their searches such as date, publisher etc	1	Low	Akash
Sprint-3	Chatbot	USN-8	A functional chatbot can respond to user queries in real time	2	Medium	Venkatasrinivas,Murali Dharan
Sprint-4	Server	USN-9	Administrator verifies and validates the news available in the database	3	High	Jayant,Sanjay,Venkatasrinivas
		USN-10	Administrator provides news articles with video and audio content	3	High	Akash,Venkatasrinivas,Murali Dharan

6.2 Sprint Delivery Schedule:

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	8 Days	29 Oct 2022	03 Oct 2022	12	05 November 2022
Sprint-2	20	4 Days	05 Oct 2022	09 Nov 2022	3	9 November 2022
Sprint-3	20	3 Days	09 Nov 2022	12 Nov 2022	2	12 November 2022
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022	3	19 November 2022

6.3 Report from JIRA:
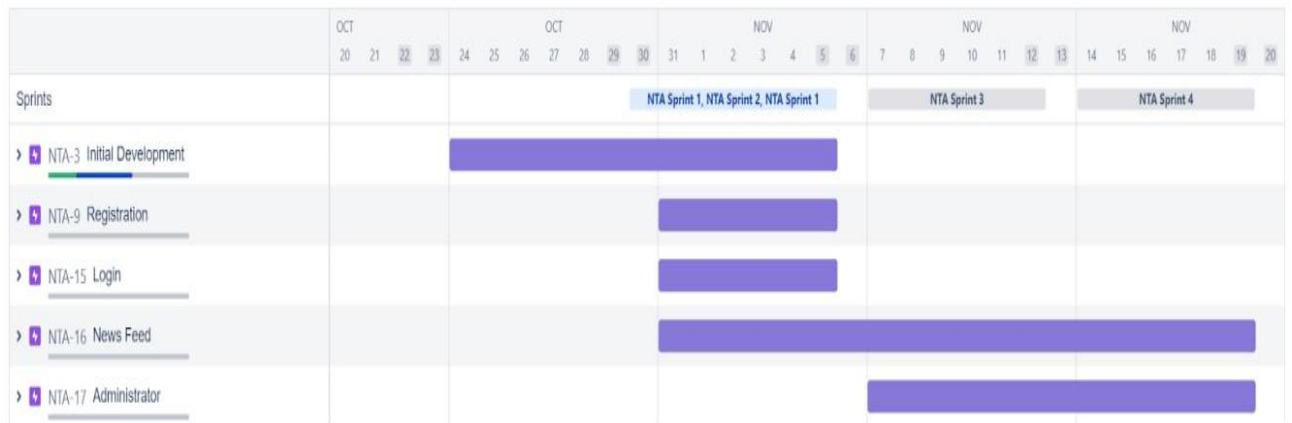
7. CODING & SOLUTIONING

7.1 Feature 1 - Personalization
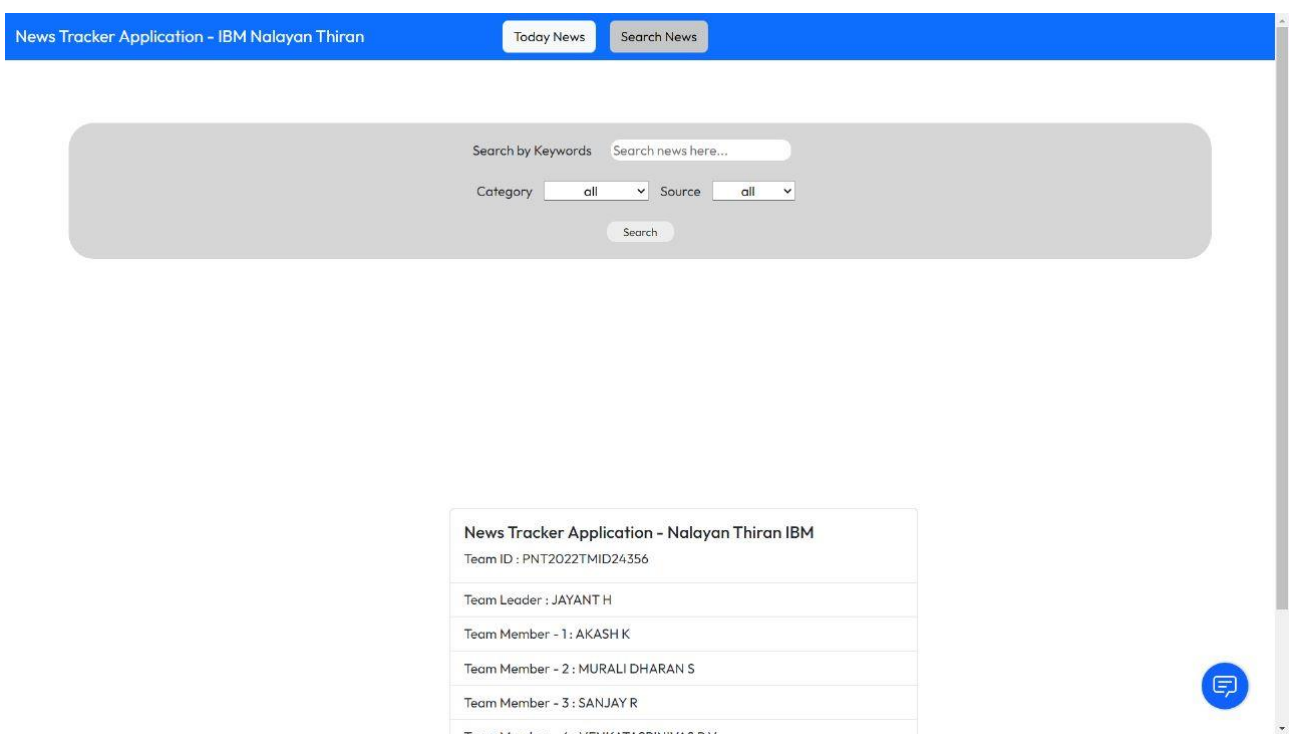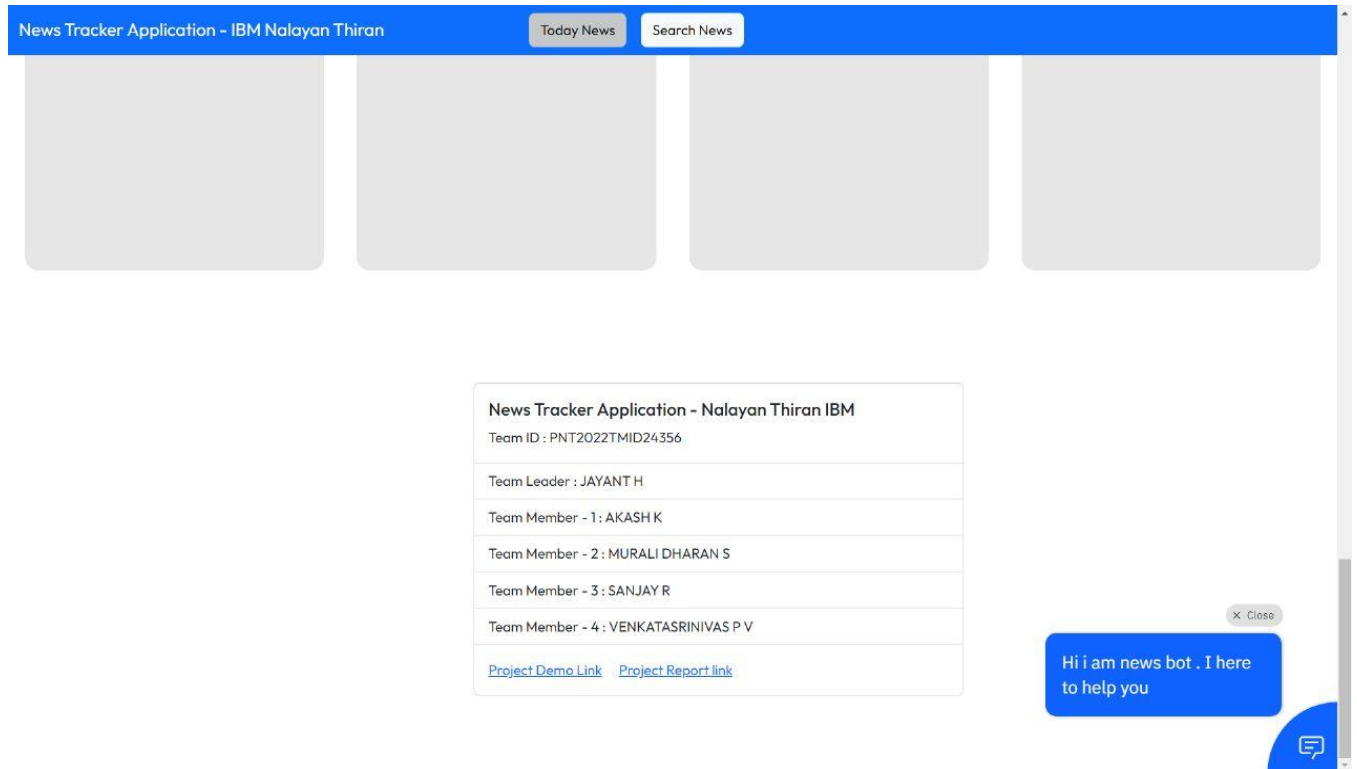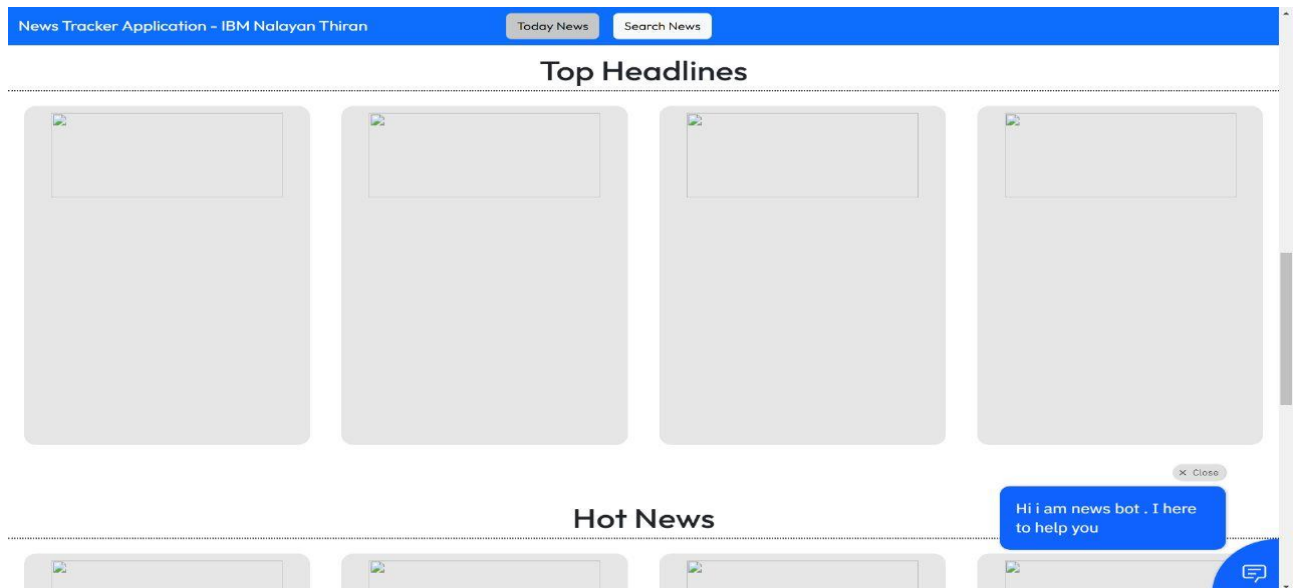
The "set-preferences" allows registered and logged in users (who haven't already set their preferences) to set them up.

This allows the user to read articles based on their preference. The users is made to choose 5 topics of their choice from which the latest articles will be displayed after they login. This gives the user a sense of personalisation and encourages them to visit the application more often.

7.2 Feature 2 – Search

The search bar allows users to search topics, sources and authors of their choice. The newsApi will fetch the articles based on the query and display them as tiles.

3. Database Schema

PREFERENCES

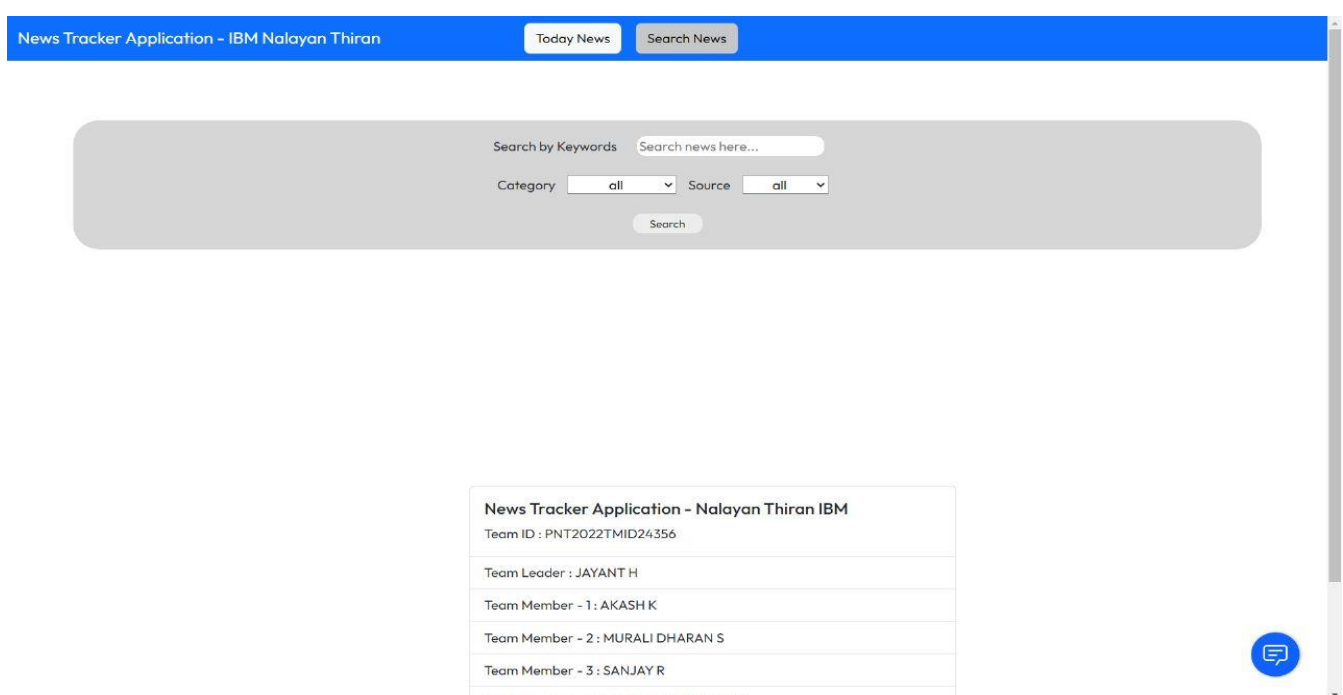No statistics available.

Name	Data type	Nullable	Length	Scale	
EMAIL	LONG VARCHAR	N	32700	0	👁
PREFERENCE_1	LONG VARCHAR	Y	32700	0	👁
PREFERENCE_2	LONG VARCHAR	Y	32700	0	👁
PREFERENCE_3	LONG VARCHAR	Y	32700	0	👁
PREFERENCE_4	LONG VARCHAR	Y	32700	0	👁
PREFERENCE_5	LONG VARCHAR	Y	32700	0	👁

USERS

No statistics available.

Name	Data type	Nullable	Length	Scale	
EMAIL	LONG VARCHAR	N	32700	0	👁
USERNAME	LONG VARCHAR	N	32700	0	👁
USERPASSWORD	LONG VARCHAR	N	32700	0	👁

The Users table is used for storing and retrieving user data when they register and login respectively. Users on registering their emails and usernames will be able to track their data throughout the application

The Preferences table is used for setting the user's topics of choice for the landing page. Every user is asked to set 5 topics on their first login and these preferences are set for the user throughout till the user's data is removed.

8. TESTING

8.1 Test Cases

The Test cases for the News Tracker application are as follows

- Verify If user can Sign up to the account

- Verify If already signed up user cannot log into the account

- Verify if user is able to see Login/Register when clicked on it
- Verify if user is able to filter articles based on categories

- Verify if user is able to see detailed information when clicked on read more

8.2 User Acceptance Testing

Test_Case ID	Feature Type	Component	Test Scenario	Steps to Execute	Test Data	Expected Result	Actual Result	Status
SignUpPage_TC1	Functional	Home Page	Verify If user can Sign up to the account	1. Enter URL 2. In Navigation bar, click on Login/Register 3. Enter Credentials like Email, Username	Mail: testing123@gmail.com Username: Ram_Haridhra Password: password	Logged in messge should be displayed	Working as Expected	Pass
SignUpPage_TC2	Functional	Home Page	Verify If already signed up user cannot log into the account	1. Enter URL 2. In Navigation bar, click on Login/Register 3. Enter Credentials like Email, Username	Mail: testing123@gmail.com Username: Ram_Haridhra Password: password	User already exists, please log in	Working as Expected	Pass
SignInPage_TC3	UI	Home Page	Verify if user is able to see Login/Register when clicked on it	1. Enter URL 2. In Navigation bar, click on Login/Register 3. Verify SignIn/SignUp Displayed	http://127.0.0.1:5000/	SignIn/SignUp Page should be displayed	Working as Expected	Pass
SignUpPage_TC4	UI	Home Page	Verify if user is able to see Login/Register when clicked on it	1. Enter URL 2. In Navigation bar, click on Login/Register 3. Verify SignIn/SignUp Displayed	http://127.0.0.1:5000/	SignIn/SignUp Page should be displayed	Working as Expected	Pass
CategoriesPage_TC5	Functional	Categories	Verify if user is able to filter articles based on categories	1. Enter URL 2. In Navigation bar, click on Categories 3. Verify if filter is applied	http://127.0.0.1:5000/	Articles based on categories should be displayed	Working as Expected	Pass
ArticlesPage_TC6	Functional	Articles Page	Verify if user is able to see detailed information when clicked on read more	1. Enter URL 2. Go to Articles Page, click on Read more 3. Verify if detailed information is shown	http://127.0.0.1:5000/	Detailed Article should be displayed when clicked on read more	Working as Expected	Pass

9. RESULTS

9.1 Performance Metrics

Latest Performance Report for:
http://159.122.178.45:31536/

Report generated: Fri, Nov 18, 2022 10:30 AM -0800
Test Server Location: 🇨🇦 Vancouver, Canada
Using: 🔴 Chrome (Desktop) 103.0.5060.134, Lighthouse 9.6.4

GTmetrix Grade ?

B	Performance ? 88%	Structure ? 62%

Web Vitals ?

LCP ? 1.5s	TBT ? 92ms	CLS ? 0

Performance Metrics

The following metrics are generated using Lighthouse Performance data.

Metric details OFF

First Contentful Paint ?	Longer than recommended 1.5s	Time to Interactive ?	Good - Nothing to do here 2.3s
Speed Index ?	OK, but consider improvement 1.6s	Total Blocking Time ?	Good - Nothing to do here 92ms
Largest Contentful Paint ?	OK, but consider improvement 1.5s	Cumulative Layout Shift ?	Good - Nothing to do here 0

Speed Visualization ?

0.3s	0.6s	0.9s	1.2s	1.5s	1.8s	2.1s	2.4s	2.7s	3s

TTFB: 587ms
Redirect: 0ms
Connect: 155ms
Backend: 432ms

First Contentful Paint: 1.5s
Largest Contentful Paint: 1.5s

Time to Interactive: 2.3s
Onload Time: 2.3s

Fully Loaded Time: 3.0s
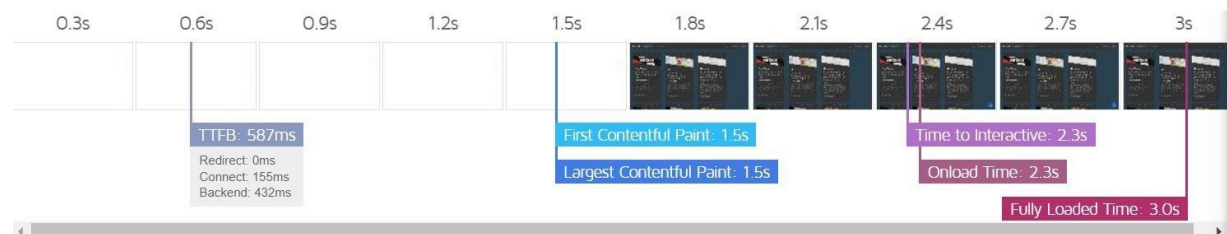
10. ADVANTAGES & DISADVANTAGES

Advantages

- The user's are able to keep track of their favourite sources, categories or authors with a few clicks

- The search bar allows the user to query results based on a large number of options and combinations

- Chatbot integration for assistance in account creation and preference setting

- Multiple categories and endless articles fetched using newsapi

Disadvantages

- Since the news articles are fetch from an external API, the validity of the data cannot be guaranteed

- Users have to give accurate keywords to search for results

11. CONCLUSION

With the help of the internet and external application programming interfaces, it is possible to keep the users informed of the current affairs. With a vast database of articles and references, users are even encouraged to read more.

As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent. Users' demands and requirements for personalisation increases day by day. The application's personalisation feature will attract users to spend more time browsing through articles.

And finally, the search bar, will help readers located certain articles based on multiple filters like date, source, language and author to give the users a better user experience.

12. FUTURE SCOPE

- Provide filter options for user's to track articles

- Modification and deletion of preferences

- Location based rendering of articles

- Allowing users to report falsified and misleading information

13.APPENDIX

Source Code:

main.py:

```python
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='2f8ecf96502842f284ab864d0a0ba76b ')


@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)
```

```python
        # 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:
```

```python
            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

```javascript
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);
```

```javascript
        }
    }

    // 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].valu
e);
    };

    // 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();
        }
```

```javascript
    };

    // define reset onlick 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
```

```javascript
(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);

            });
        }
    )
```

```javascript
        .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");
```

```javascript
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);

```javascript
  // 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

= `<b>Author:</b> ${article.author}`;

text_div.appendChild(author_p);
```

```javascript
    // add source info     let source_p =
document.createElement("p");     source_p.innerHTML =
`<b>Source:</b> ${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 = `<b>Date:</b>
${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);`
```

```javascript
    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 =
```

```javascript
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

```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>

<br>
```

```html
<br>

<br>

  <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>
```

```html
        </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">
```

```html
<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'>
```

```html
          <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'>
```

```html
            <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>
```

```html
                    <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>

      </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">
```

```html
      <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>
```

Index.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");
```

```javascript
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
```

```javascript
    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 onlick 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);
    })
```

```javascript
}

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() {
```

```javascript
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 => {
```

```javascript
                display_search_results(data);

            });

        }

    )

    .catch(err => {

        console.log(err);

    })

}


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;
```

```javascript
    // 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
```

```javascript
    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);
```

```javascript
    }
    // 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);
    }
}


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]);
    }
}
```

```javascript
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;
```

```javascript
}


// 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);
```

```javascript
        // 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");
```

```javascript
                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";

            }

        }

}


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 = `<b>Author:</b> ${article.author}`;
```

```javascript
text_div.appendChild(author_p);


// add source info

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

source_p.innerHTML = `<b>Source:</b> ${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 = `<b>Date:</b> ${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";
```

```javascript
        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");
```

```javascript
    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");
```

```javascript
    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;

}
```

```javascript
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;
```

style.css

```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%;
```

```css
    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;
```

```css
}
.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;
}
```

```css
#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;
```

```css
    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;
```

```css
    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;
```

```css
    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;

}
```

```css
#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;
```

```css
    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;

}
```

```css
#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;

}
```

```css
.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 Link: https://github.com/IBM-EPBL/IBM-Project-53979-1661586582