

# **PROJECT REPORT**

## **NEWS TRACKER APPLICATION**

### **1. INTRODUCTION**

#### **1.1 Project Overview**

The main objective of the project is to provide people a web application through which people can access all types of news and information. Through this application, any user can gain technical knowledge of the world and its surrounding with just one click ahead. User does not have to visit multiple sites for different related information. All information is going to be in one place.

#### **1.2 Purpose**

The purpose is to develop an web application, which will eliminate the problems faced in the current scenario. This application will provide all the information and news related to E-sports, science, and Technology or that are in trend at one place. So, it will save time and efforts of the users by making it more efficient. Using, this application will terminate the possibility of information redundancy.

### **2. LITERATURE SURVEY**

#### **2.1 Existing Problem**

Our research utilizes the current trends and technology to design and develop a web application for every user. News organizations whose web applications only provide users with their articles or videos are missing a big opportunity. There are multiple news sharing apps used by a single user and are often spammed with notifications. There is also a lot of fake news.

#### **2.2 Reference**

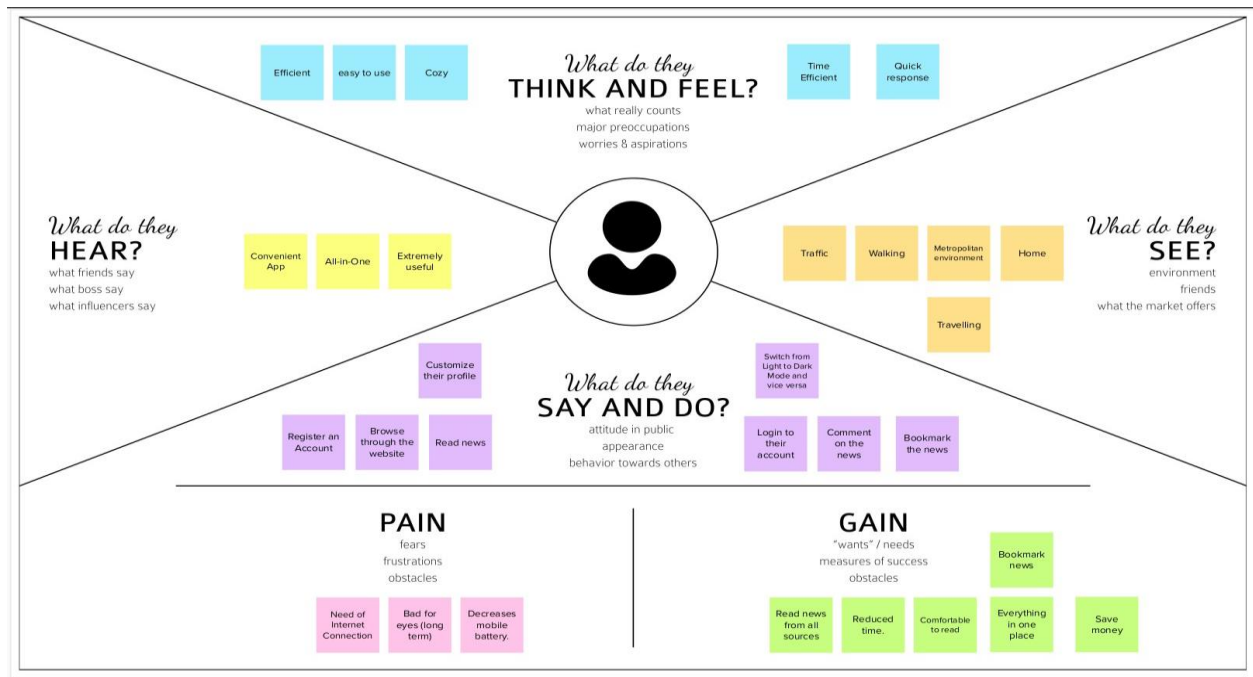
1. Exploring mobile news reading interactions for news app personalisation – Marios Constantine & John Dowell.
2. Detection and Tracking in News Articles – Sagar Patel, Sanket Suthar.
3. Following the feed with a news tracker – Michael William McCracken

#### **2.3 Problem Statement**

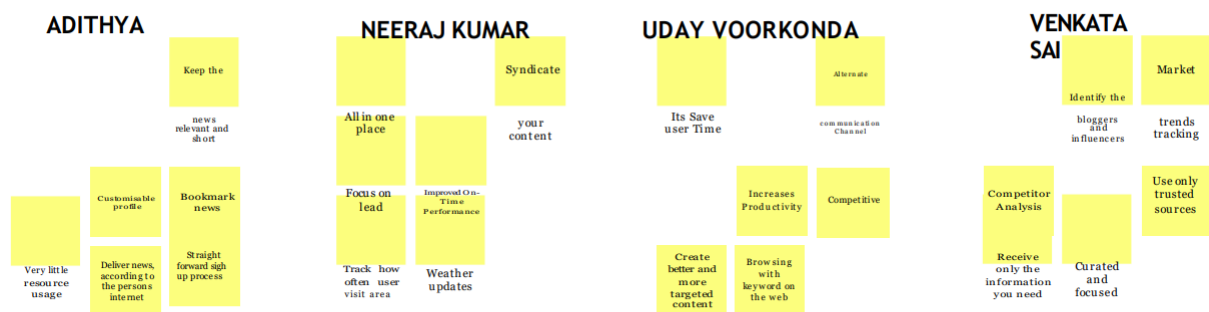
The customer needs a way to get relevant news based on his choices so that the customer does not have to spend a lot of time on searching news. News is filled with ads and spams annoys and irritates the customer and affects the user experience. Market is full of news with all categories. Customers are 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.

### 3. IDEATION AND PROPOSED SOLUTION

#### 3.1 Empathy map canvas



#### 3.2 Ideation & Brainstorming

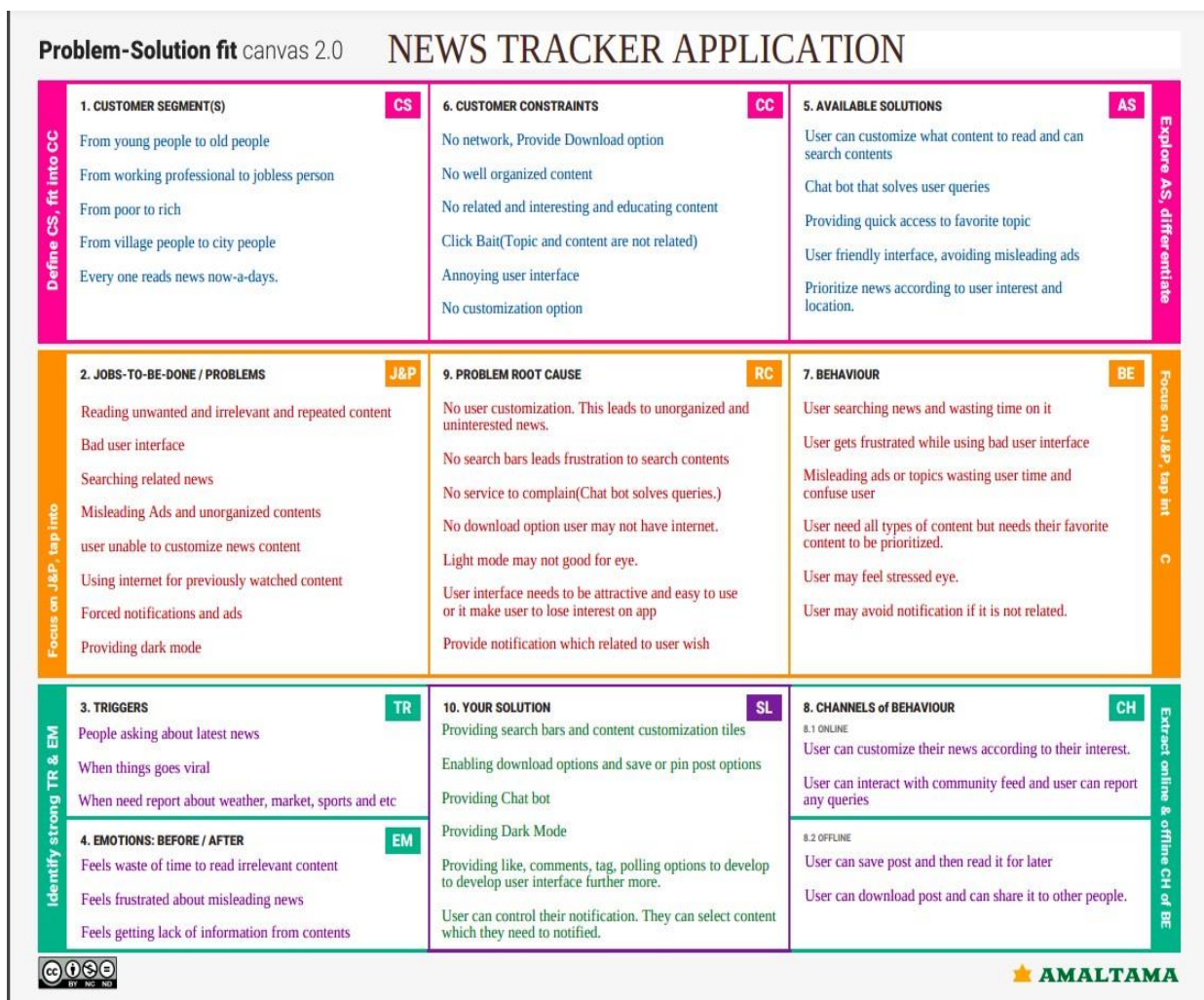


### 3.3 Proposed Solution

As a team, we realized that we have different skill sets and felt that we would all be able to cater to this problem statement in a very strategic and collaborative manner.

Plus it was interesting to see how all of us had a very different take on news consumption. A news-sharing app wants to help users find relevant and important news easily every day and also understand explicitly that the news is not fake but from proper sources

### 3.4 Problem Solution fit



## 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	Registration through Gmail
FR-2	User Confirmation	Confirmation via Google Auth
FR-3	User Login	User login with username and password Login with Gmail
FR-4	User Interface	User goes through the filtered news provided by application
FR-5	User Account Settings	User can update account credentials, update interests and choices of news and report any issues

### 4.2 Non-Functional Requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<p>The application is easily navigable, it has an interface which is user-friendly. The login/register page has simple terms which can be filled by a user who has not done it before. The user interface navigates the user to the Home page and asks for interests.</p> <p>The settings can be quickly accessed by the user by going to the account settings, where the credentials and interests can be changed anytime.</p>

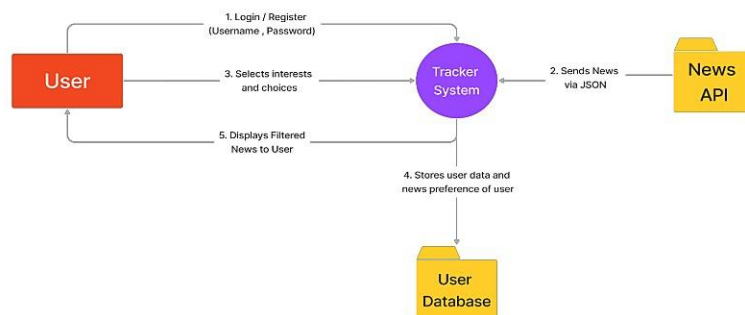
NFR-2	<b>Security</b>	The application uses Google OAuth 2.0 / Auth0 for User authentication.
NFR-3	<b>Reliability</b>	The application should be able to perform without failure in 95 percent of all use cases during a month since the IBM cloud doesn't face down time so often.
NFR-4	<b>Performance</b>	The application's news feed page has a loading time of 6 seconds.
NFR-5	<b>Availability</b>	The application must be available to the users at least 99% of the time in a month.
NFR-6	<b>Scalability</b>	The application should be able to handle upto 10,000 users at a time considering the cloud factors.

## 5. PROJECT DESIGN

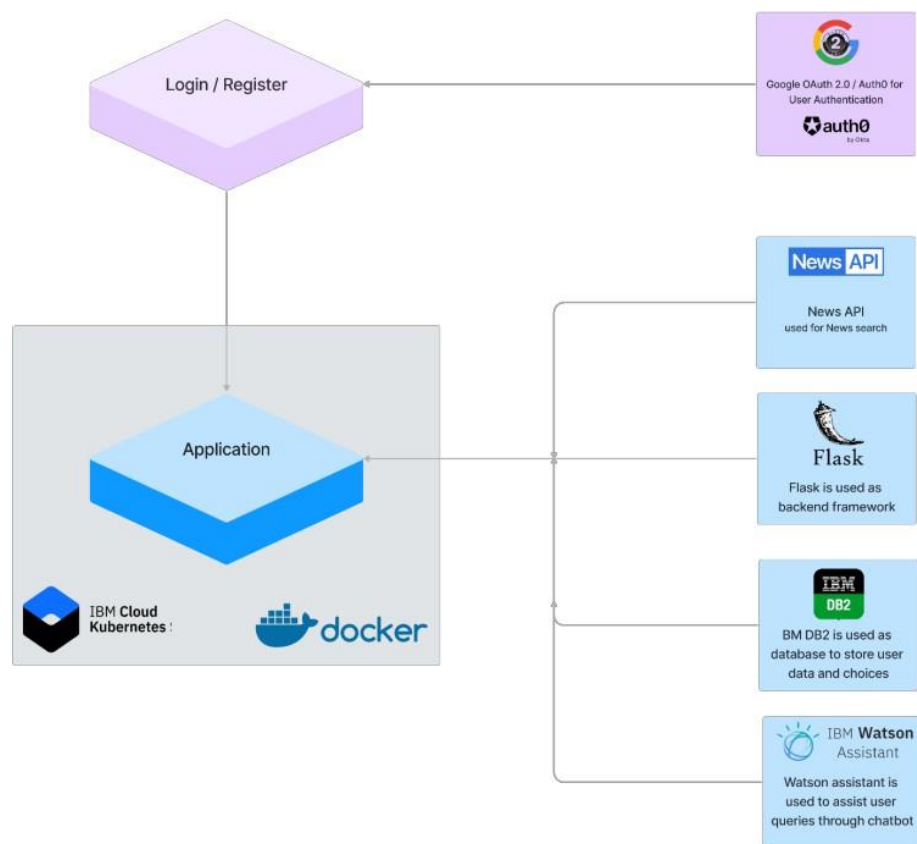
### 5.1 Data Flow Diagrams

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

LEVEL 0 DATAFLOW 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 user / Web 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	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 and 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 log out 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 choice in account settings.		Medium	Sprint-4
Customer Care Executive	Chat Bot / Query Section	USN-9	Solve issues brought up by client.		Medium	Sprint-4
Administrator		USN-10	Roll out updates and bug fixes.		High	Sprint-4

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	1	2	1	9
Duplicate	1	0	0	0	1
External	1	0	0	1	2
Fixed	7	1	2	2	12
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	14	2	4	4	24



## 6.2 Sprint Delivery Schedule

Section	Total Cases	Not Tested	Fail	Pass
Login Page	4	0	0	4
Registration Page	1	0	0	1
Home Page	5	0	0	5

## 7. CODING & SOLUTIONING

### 7.1 FEATURE 1 – Search Filter

```
SEARCH FILTER

@app.route("/search", methods=['POST', 'GET'])
def search_news():
    search_query = request.form['search']
    search_results = newsapi.get_everything(q=search_query)
    return render_template('search.html', session=session.get('user'),
        search_results=search_results, categories=all_categories,
        search_query=search_query)
```

### 7.2 FEATURE 2 - Chat-bot

```
CHAT-BOT

window.watsonAssistantChatOptions = {
    integrationID: "1713fc16-946a-4bc2-ad0a-52f55338a079", // The ID of this
integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "fc27de01-58f5-4803-9744-ec8799654c59", // The ID of
your service instance.
    onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/"
+ (window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
});
```



## 7.3 Category Filter

```
● ● ● CATEGORY FILTER

@app.route("/business")
def get_business():
    cat_news = newsapi.get_top_headlines(
        category='business',
        language='en',
        page_size=50)
    return render_template('individual_category.html', session=session.get('user'),
        articles=cat_news, article_category='business', categories=all_categories)

@app.route("/entertainment")
def get_entertainment():
    cat_news = newsapi.get_top_headlines(
        category='entertainment',
        language='en')
    return render_template('individual_category.html', session=session.get('user'),
        articles=cat_news, article_category='entertainment', categories=all_categories)

@app.route("/general")
def get_general():
    cat_news = newsapi.get_top_headlines(
        category='general',
        language='en')

    # return cat_news
    return render_template('individual_category.html', session=session.get('user'),
        articles=cat_news, article_category='general', categories=all_categories)

@app.route("/health")
def get_health():
    cat_news = newsapi.get_top_headlines(
        category='health',
        language='en')

    # return cat_news
    return render_template('individual_category.html', session=session.get('user'),
        articles=cat_news, article_category='health', categories=all_categories)
```

## 8. TESTING

### 8.1 Test Cases

				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requlite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
loginPage_TC_001	UI	Login Page	Verify user is able to see the login/signup popup when goes to the App's URL.		1.Enter App's URL and click go 2.Verify login/signup popup displayed or not	<a href="http://169.51.207.49:31362/">http://169.51.207.49:31362/</a>	login/signup Block should display	Working as expected	Pass				Ram Bharath V, Vignesh R
loginPage_TC_002	Functional	Login Page	Verify the UI elements in login/signup Block		1.Enter App's URL and click go 2.Verify login/signup popup with below UI elements: a. Email text box b. Password text box c. Login button d. New customer? Create account link	<a href="http://169.51.207.49:31362/">http://169.51.207.49:31362/</a>	Application should show below UI elements: a.Email text box b.password text box c.Login button with orange colour d.New customer? Create account link	Working as expected	Pass				Vignesh E, Navene MS
RegisterPage_TC_003	Functional	Register page	Verify user is able to Register into application with required credentials		1.Enter App's URL( <a href="http://169.51.207.49:31362/">http://169.51.207.49:31362/</a> ) and click go 2.Click on Register button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on Register button	Username: geniousource@gmail.com password: Testing123	The user's registration details should be saved in the database and the user should be navigated to the Home section.	Working as expected	Pass				Rishik K V, Ram Bharath V
loginPage_TC_004	Functional	Login page	Verify user is able to log into application with valid credentials		1.Enter URL( <a href="http://169.51.207.49:31362/">http://169.51.207.49:31362/</a> ) and click go to login page. 2.Enter invalid username/email in Email text box. 3.Enter valid password in password text box. 4.Click on login button	Username: geniousource@gmail.com password: Testing123	The user should be redirected to the Home page where the categories and news is displayed.	Working as expected	Pass				Ram Bharath V, Rishik K V

loginPage_TC_005	Functional	Login page	Verify user is able to log into application with invalid credentials		1.Enter URL( <a href="http://169.51.207.49:31362/">http://169.51.207.49:31362/</a> ) and click go. 2.Enter Valid username/email in Email text box 3.Enter invalid password in password text box 4.Click on login button	Username: phishand@gmail.com password: Testing123@7668766876676	Application should show "Invalid email or password" validation message to the user.	Working as expected	Pass				Navene M S, Vignesh E
HomePage_TC_006	UI	Home page	Verify user is able to see the Home page after logging in to the app with valid credentials.		1. Enter Entering the app url, go to login. 2. login in to the page with credentials.	Username: geniousource@gmail.com password: Testing123	Application should show the home page and it's element.	Working as expected	Pass				Rishik K V, Ram Bharath V
HomePage_TC_007	Functional	Home page	Verify user is able to see the news and categories		1. Enter Entering the app url, go to login. 2. login in to the page with credentials.	Username: geniousource@gmail.com password: Testing123	Application should show the news and categories to the user.	Working as expected	Pass				Vignesh R, Vignesh E
HomePage_TC_008	Functional	Home page	Verify user is able to use all categories of news.		1. Enter the home page. 2. Traverse through the categories and use all the categories of news.	Categories: Business, Technology, Entertainment, Sports, General, Health	Application should show different categories of news to the user.	Working as expected	Pass				Navene M S, Vignesh R
HomePage_TC_009	Functional	Home page	Verify user is able to search news with keywords.		1. Enter the home page. 2. Click the search bar and enter the keywords the user want to search and see the news.	Keywords = iPhone, Smart, Samsung	Application should display the news related to the keywords.	Working as expected	Pass				Ram Bharath V, Vignesh E
HomePage_TC_010	Functional	Home page	Verify user is able to logout from application.		1. Enter the home page. 2. Click Logout.	Logout	Application should be redirected to the login page but the account will be logged out.	Working as expected	Pass				Ram Bharath V, Vignesh R

## 8.2 User Acceptance Testing

### 1.PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the News Tracker Application project at the time of the release to User Acceptance Testing(UAT).

### 2. DEFECT ANALYSIS

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	1	2	1	9
Duplicate	1	0	0	0	1
External	1	0	0	1	2
Fixed	7	1	2	2	12
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	14	2	4	4	24

### 3. TEST CASE ANALYSIS

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Login Page	4	0	0	4
Registration Page	1	0	0	1
Home Page	5	0	0	5

## **9. ADVANTAGES**

1. Viewers can get their news straight off their smartphone or tablet computer.
2. News is at their fingertips in an instant. An online newspaper can be read more elaborate than a printed newspaper.
3. You can read the old issues too very easily at the click of the mouse.
4. 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

## **DISADVANTAGES**

1. It CAN BE limited by time.
2. It may rely too heavily on personalities, emotions, opinions... not facts.
3. It can shortchange complex stories or avoid them altogether.

## **10. CONCLUSION**

The Motivation and scope behind this project are to connect people through this application and provide a medium to share their views on the topic/news/information. Then, People with the same interest can interact with each other. However, they can even share more information on the topic. This app while cross-checks the redundancy in the information along with the false and misleading information, which later results in panic in the people.

## **11. FUTURE SCOPE**

Location feature with automation can be implemented which means as user move from one city to other local news will change as per it. Offline Reading can be improve will more efficient way on full articles. Data quality check needed. If API can't reach to certain article source it gives null value which can cause problem in JSON parsing .

## 12. APPENDIX

### SOURCE

### CODE

```
import json
from os import environ as env
from urllib.parse import quote_plus, urlencode

from authlib.integrations.flask_client import OAuthfrom
dotenv import find_dotenv, load_dotenv
from flask import Flask, redirect, render_template, session, url_for, requestfrom

newsapi import NewsApiClient

ENV_FILE = find_dotenv()
if ENV_FILE:
    load_dotenv(ENV_FILE)

app = Flask(__name__)
# env_config = env.getenv("APP_SETTINGS", "config.DevelopmentConfig")#
app.config.from_object(env_config)
app.secret_key = env.get("APP_SECRET_KEY")

newsapi = NewsApiClient(api_key=env.get("NEWS_API_KEY"))

oauth = OAuth(app)

oauth.register(
    "auth0",
    client_id= env.get("AUTH0_CLIENT_ID"),
    client_secret= env.get("AUTH0_CLIENT_SECRET"),
    client_kwargs={
        "scope": "openid profile email",
    },
    server_metadata_url=f'https://{env.get("AUTH0_DOMAIN")}/.well-known/openid-
configuration'
)
```

```
all_categories = ["business", "entertainment", "general", "health", "science", "sports",  
"technology"]
```

```
# @app.route("/")#
```

```
def index():
```

```
# sample_config_data = app.config.get("SAMPLE")#  
    return sample_config_data
```

```
@app.route("/login") def
```

```
login():
```

```
    return oauth.auth0.authorize_redirect(  
        #should be redirect_uri and not redirect_url  
        redirect_uri=url_for("callback", _external=True)  
    )
```

```
@app.route("/callback", methods=["GET", "POST"])def
```

```
callback():
```

```
    token = oauth.auth0.authorize_access_token()  
    session["user"] = token  
    return redirect("/")
```

```
@app.route("/logout") def
```

```
logout():
```

```
    session.clear()  
    return redirect(  
        "https://" + env.get("AUTH0_DOMAIN")  
        + "/v2/logout?"  
        + urlencode(  
            {  
                "returnTo": url_for("home", _external=True), "client_id":  
                env.get("AUTH0_CLIENT_ID"),  
            },  
            quote_via=quote_plus,  
        )  
    )
```

```
@app.route("/") def
```

```
home():
```

```

# all_articles = get_news()
all_articles = get_top_headlines_today()
return render_template(
    # "home.html",
    "home_og.html",
    session=session.get('user'), all_articles =
    all_articles, categories= all_categories
    # pretty=json.dumps(session.get('user'),indent=4)
)

@app.route("/news") def
get_news():
    all_articles = newsapi.get_everything(q='bitcoin',
                                         sources='bbc-news,the-verge',
                                         domains='bbc.co.uk,techcrunch.com',
                                         from_param='2022-10-10',
                                         to='2022-10-12',
                                         language='en',
                                         sort_by='relevancy',
                                         page=1)

    return all_articles

# @app.route("/ind-cat", methods=["POST", "GET"])#
def get_individual_category_news():

#     category_news = newsapi.get_everything(category)

@app.route("/top-headlines-today")def
get_top_headlines_today():
    user_categories = ["business", "entertainment", "general", "health", "science", "sports",
"technology"]
    top_headlines = { }
    for i in range(len(user_categories)): top_headlines[user_categories[i]] =
        newsapi.get_top_headlines(
            # sources="",
            category=user_categories[i],

```



```

                                language='en',
                                page_size=5)
    # top_headlines.append(newsapi.get_top_headlines(#
                                category='business',
    #                                language='en'))
    return top_headlines

@app.route("/user-info")def
user_info():
    return session.get('user')

@app.route("/news-from-sources") def
get_news_from_sources():
    return

@app.route("/search", methods=['POST', 'GET'])def
search_news():
    search_query = request.form['search']
    search_results = newsapi.get_everything(q=search_query)#
    return search_results
    return render_template('search.html', session=session.get('user'),
search_results=search_results, categories=all_categories, search_query=search_query)
@app.route("/business")
def get_business():
    cat_news = newsapi.get_top_headlines(
                                category='business',
                                language='en',
                                page_size=50)

    return render_template('individual_category.html',session=session.get('user'), articles=cat_news,
article_category='business', categories=all_categories)

@app.route("/entertainment") def
get_entertainment():
    cat_news = newsapi.get_top_headlines(
                                category='entertainment',
                                language='en')
    return render_template('individual_category.html',session=session.get('user'),

```

```
articles=cat_news, article_category='entertainment', categories=all_categories)
```

```
@app.route("/general")
```

```
def get_general():
```

```
    cat_news = newsapi.get_top_headlines(  
        category='general',language='en')
```

```
    # return cat_news
```

```
    return render_template('individual_category.html',session=session.get('user'), articles=cat_news,  
article_category='general', categories=all_categories)
```

```
@app.route("/health") def
```

```
get_health():
```

```
    cat_news = newsapi.get_top_headlines(  
        category='health',language='en')
```

```
    # return cat_news
```

```
    return render_template('individual_category.html',session=session.get('user'), articles=cat_news,  
article_category='health', categories=all_categories)
```

```
@app.route("/science") def
```

```
get_science():
```

```
    cat_news = newsapi.get_top_headlines(  
        category='science',language='en')
```

```
    # return cat_news
```

```
    return render_template('individual_category.html',session=session.get('user'), articles=cat_news,  
article_category='science', categories=all_categories)
```

```
@app.route("/sports") def
```

```
get_sports():
```

```
    cat_news = newsapi.get_top_headlines(  
        category='sports',language='en')
```

```
    # return render_template(cat_news)
```

```
    return render_template('individual_category.html',session=session.get('user'), articles=cat_news,  
article_category='sports', categories=all_categories)
```

```
@app.route("/technology") def
get_technology():
    cat_news = newsapi.get_top_headlines(
        category='technology',
        language='en')
    # return cat_news
    return render_template('individual_category.html', session=session.get('user'), articles=cat_news,
        article_category='technology', categories=all_categories)

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

**GITHUB LINK:** <https://github.com/IBM-EPBL/IBM-Project-33785-1660226768>

