PROJECT REPORT

NEWS TRACKER APPLICATION

1. INTRODUCTION

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 withjust one click ahead. User does not have to visit multiple sites for different related information. All information is going to be in one place.

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, sciennce, 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

Existing Problem

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

Reference

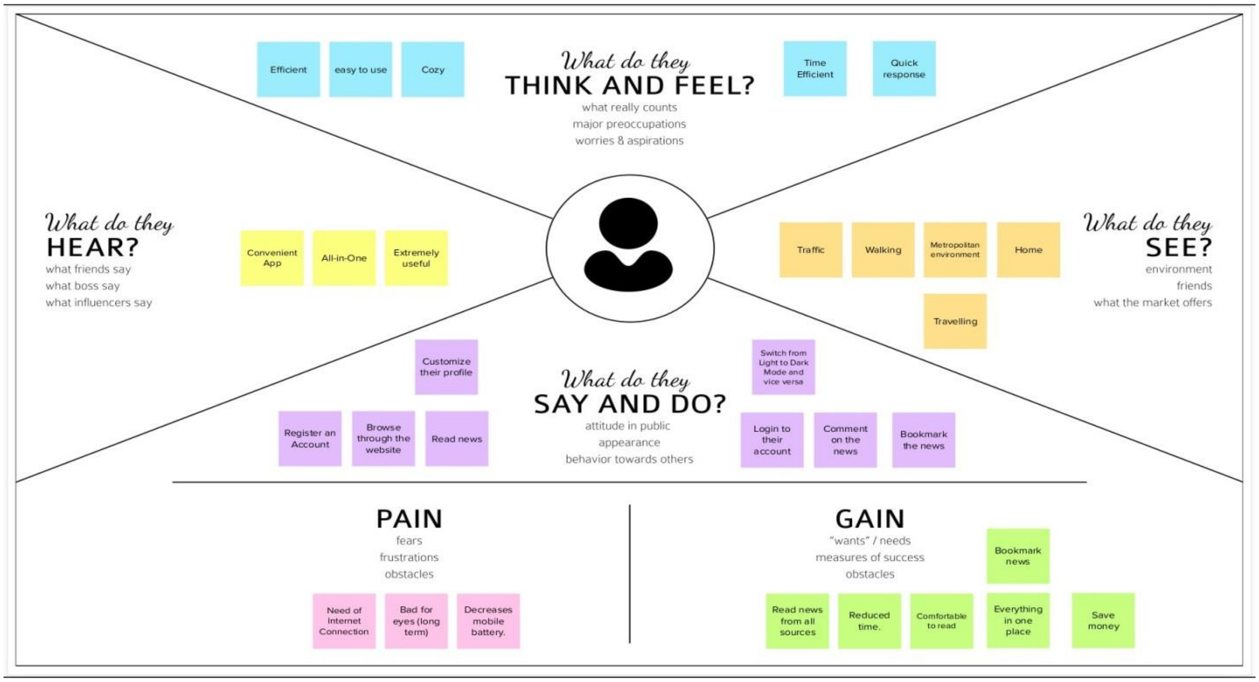
1. Exploring mobile news reading interactions for news app personalisation – Marios Constantinde & John Dowell.
2. Detection and Tracking in News Articles – Sagar Patel, Sanket suthar.
3. Following the fed with a news tracker – Michael william Mc Cracken

Problem Statement

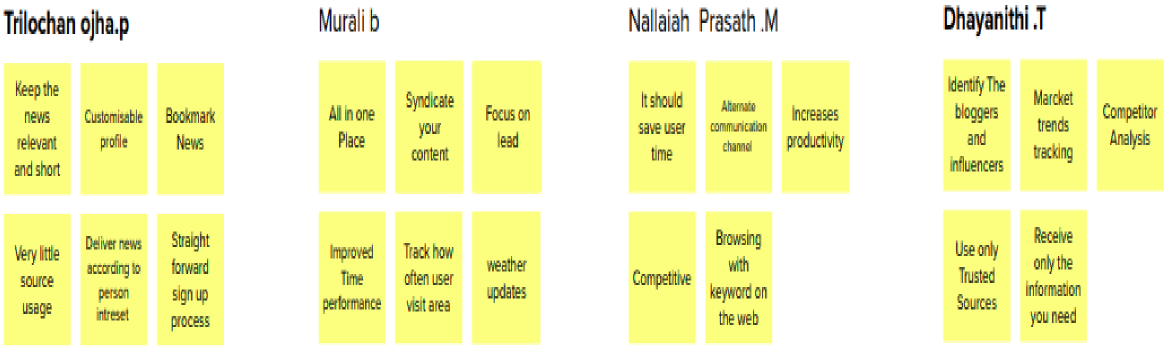
The customer needs a way to get relevant news based on his choices so that thecustomer does not have to spend a lot of time on searching news.News is filled with ads and spams annoys and irritates the customer and affects the user experience.Market is full of news with all categories. Customers are not interested in allthe categories and will be more interested with their personal choices.Traditional way oftracking news is slow and obsolete, user needs a new innovative application to track news with personal based choices.

3. IDEATION AND PROPOSED SOLUTION

Empathy map canvas



Ideation & Brainstorming

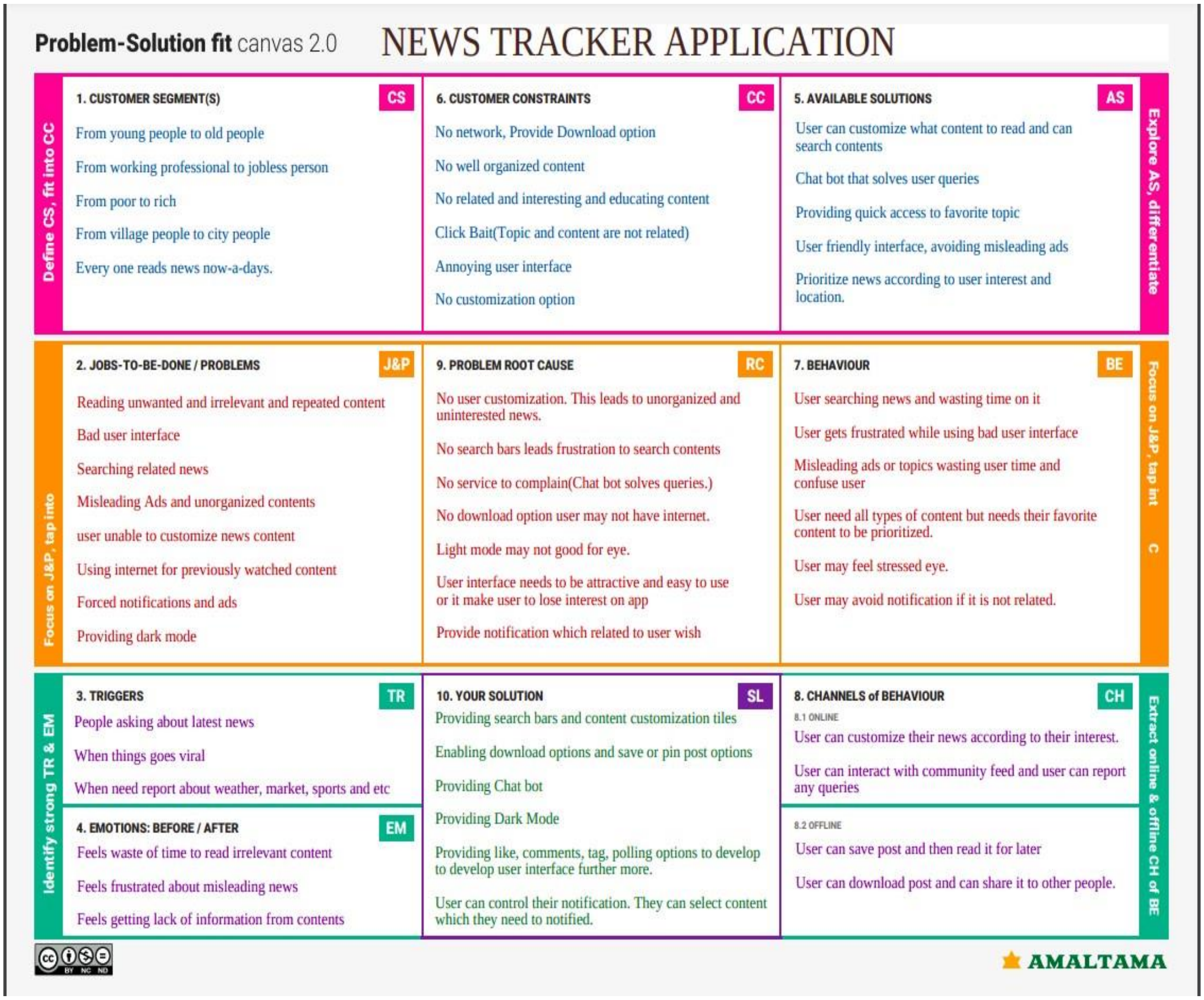


Proposed Solution

As a team, we realized that we have different skill sets and felt that we would all beable 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 newseasily every day and also understand explicitly that the news is not fake but from propersources

Problem Solution fit



Problem-Solution fit canvas 2.0 NEWS TRACKER APPLICATION

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

From young people to old people

From working professional to jobless person

From poor to rich

From village people to city people

Every one reads news now-a-days.

6. CUSTOMER CONSTRAINTS CC

No network, Provide Download option

No well organized content

No related and interesting and educating content

Click Bait(Topic and content are not related)

Annoying user interface

No customization option

5. AVAILABLE SOLUTIONS AS

User can customize what content to read and can search contents

Chat bot that solves user queries

Providing quick access to favorite topic

User friendly interface, avoiding misleading ads

Prioritize news according to user interest and location.

Explore AS, differentiate

Focus on J&P, tap into

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Reading unwanted and irrelevant and repeated content

Bad user interface

Searching related news

Misleading Ads and unorganized contents

user unable to customize news content

Using internet for previously watched content

Forced notifications and ads

Providing dark mode

9. PROBLEM ROOT CAUSE RC

No user customization. This leads to unorganized and uninterested news.

No search bars leads frustration to search contents

No service to complain(Chat bot solves queries.)

No download option user may not have internet.

Light mode may not good for eye.

User interface needs to be attractive and easy to use or it make user to lose interest on app

Provide notification which related to user wish

7. BEHAVIOUR BE

User searching news and wasting time on it

User gets frustrated while using bad user interface

Misleading ads or topics wasting user time and confuse user

User need all types of content but needs their favorite content to be prioritized.

User may feel stressed eye.

User may avoid notification if it is not related.

Focus on J&P, tap int C

Identify strong TR & EM

3. TRIGGERS TR

People asking about latest news

When things goes viral

When need report about weather, market, sports and etc

4. EMOTIONS: BEFORE / AFTER EM

Feels waste of time to read irrelevant content

Feels frustrated about misleading news

Feels getting lack of information from contents

10. YOUR SOLUTION SL

Providing search bars and content customization tiles

Enabling download options and save or pin post options

Providing Chat bot

Providing Dark Mode

Providing like, comments, tag, polling options to develop to develop user interface further more.

User can control their notification. They can select content which they need to notified.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

User can customize their news according to their interest.

User can interact with community feed and user can report any queries

8.2 OFFLINE

User can save post and then read it for later

User can download post and can share it to other people.

Extract online & offline CH of BE

AMALTAMA

4. REQUIREMENT ANALYSIS

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

Non-Functional Requirements

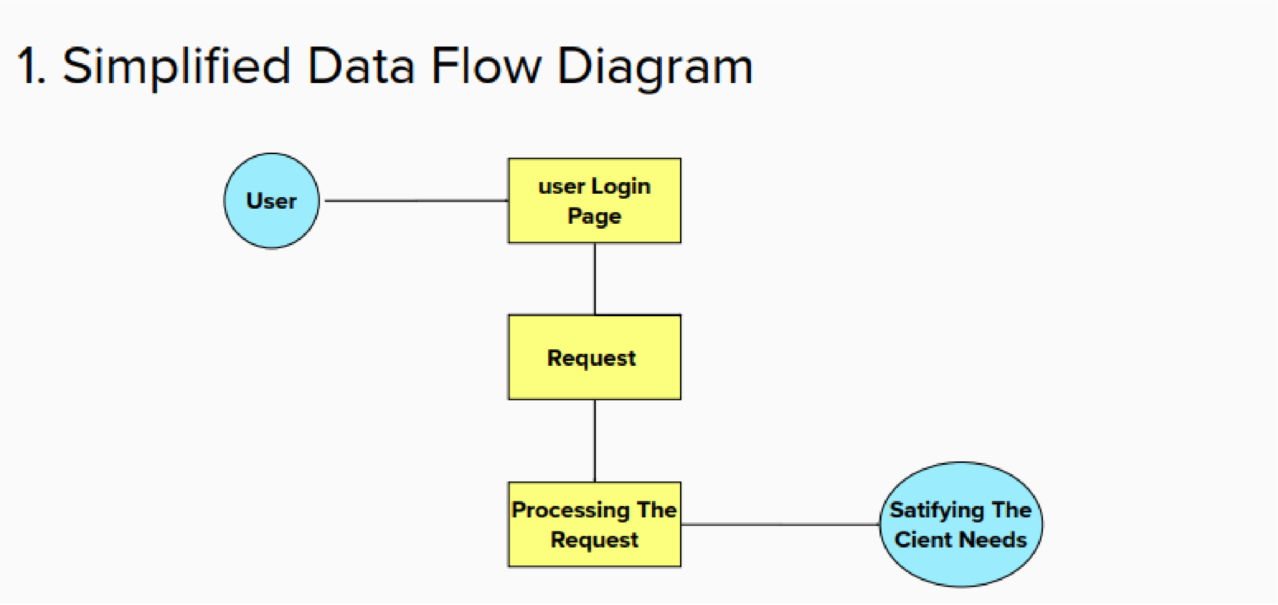
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	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. The settings can be quickly accessed by the user by going to the account settings, where the credentials and interests can be changed anytime.

NFR-2	**Security**	The application uses Google OAuth 2.0 / Auth0 for User authentication.
NFR-3	**Reliability**	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	**Performance**	The application's news feed page has a loading time of 6 seconds.
NFR-5	**Availability**	The application must be available to the users at least 99% of the time in a month.
NFR-6	**Scalability**	The application should be able to handle upto 10,000 users at a time considering the cloud factors.

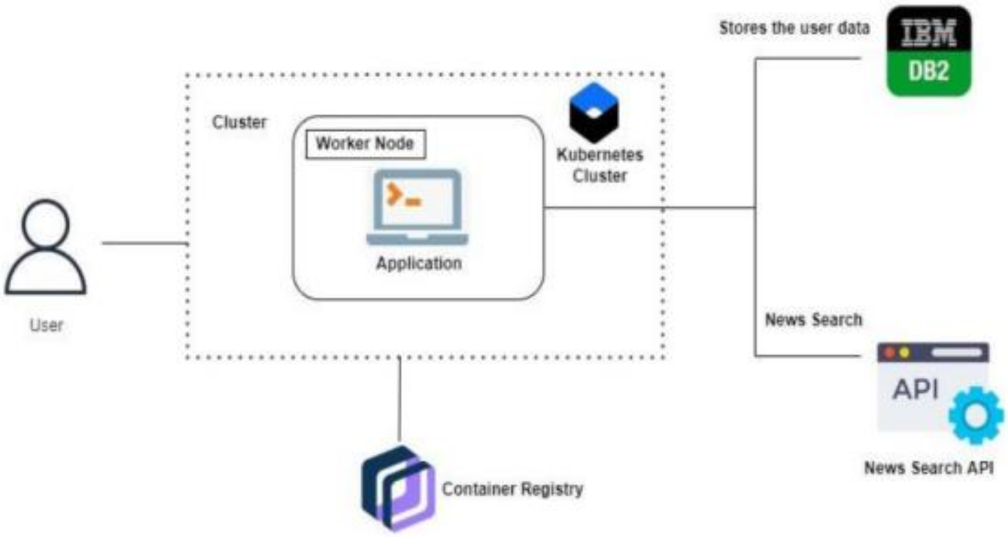
5. PROJECT DESIGN

Data Flow Diagrams

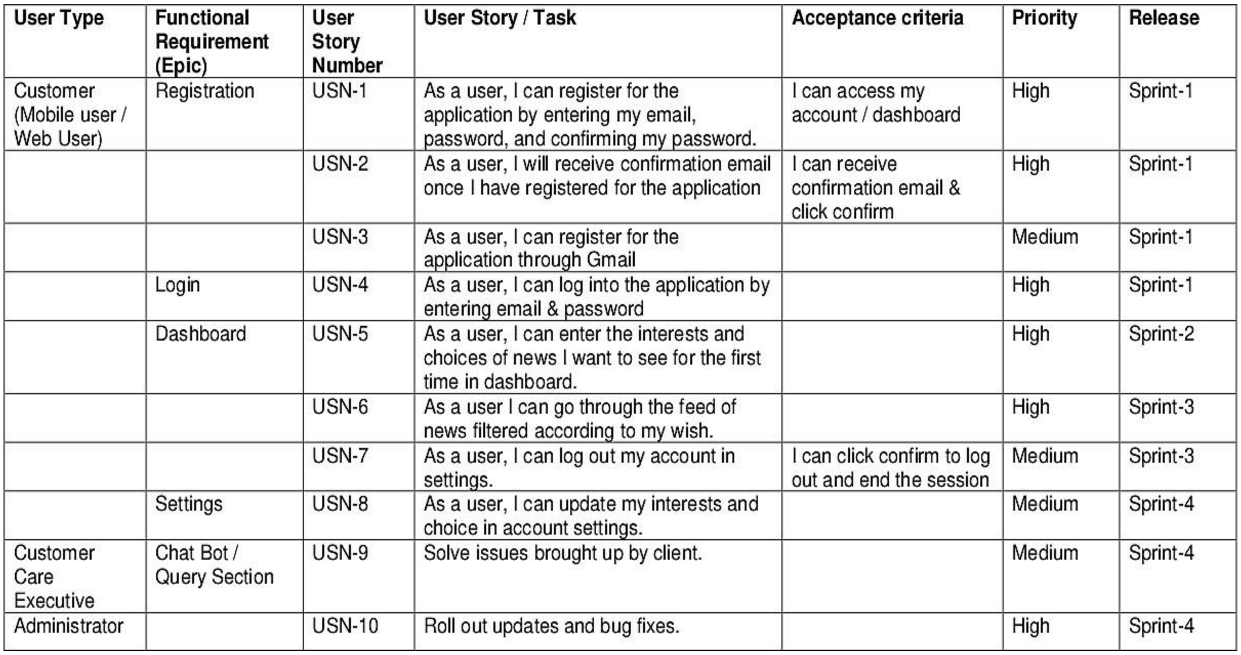


1. Simplified Data Flow Diagram

Solution & Technical Architecture



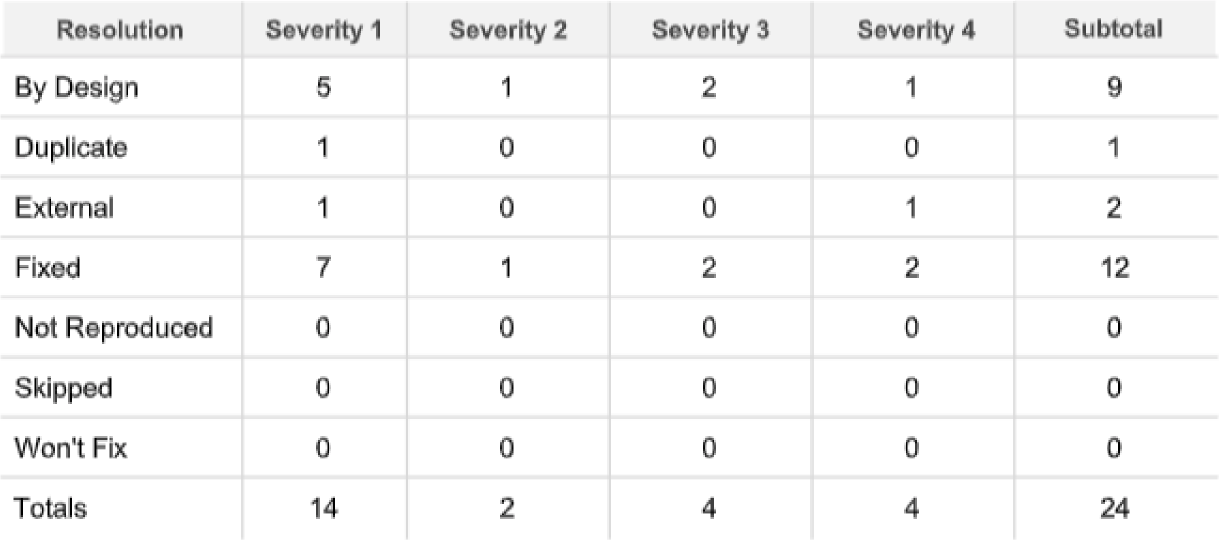
User Stories



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

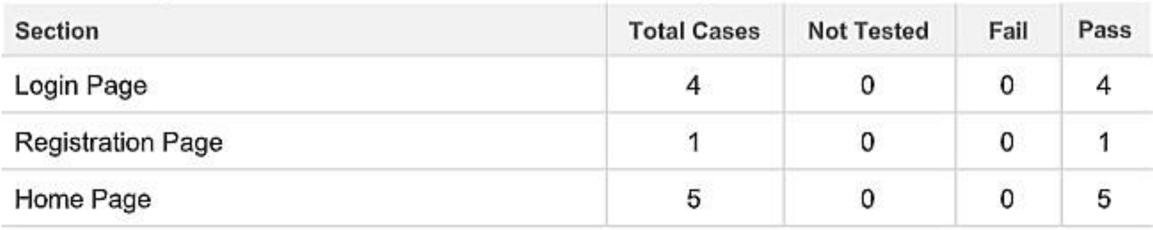
6. PROJECT PLANNING & SCHEDULING

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

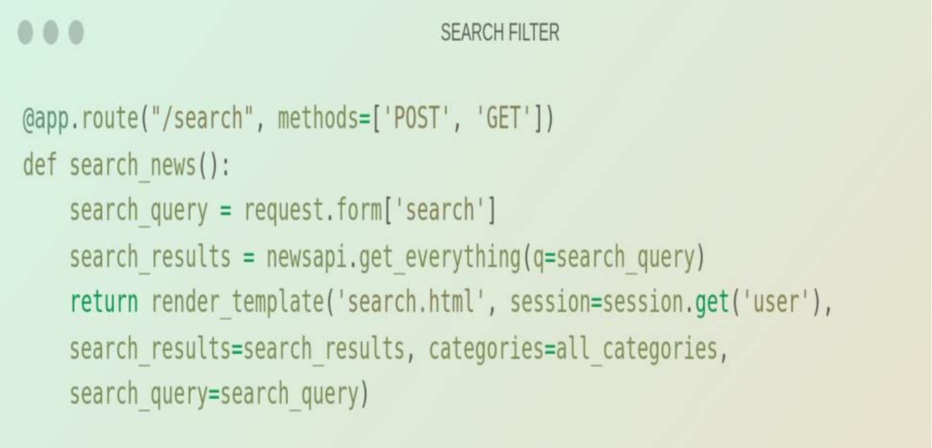
Sprint Delivery Schedule



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

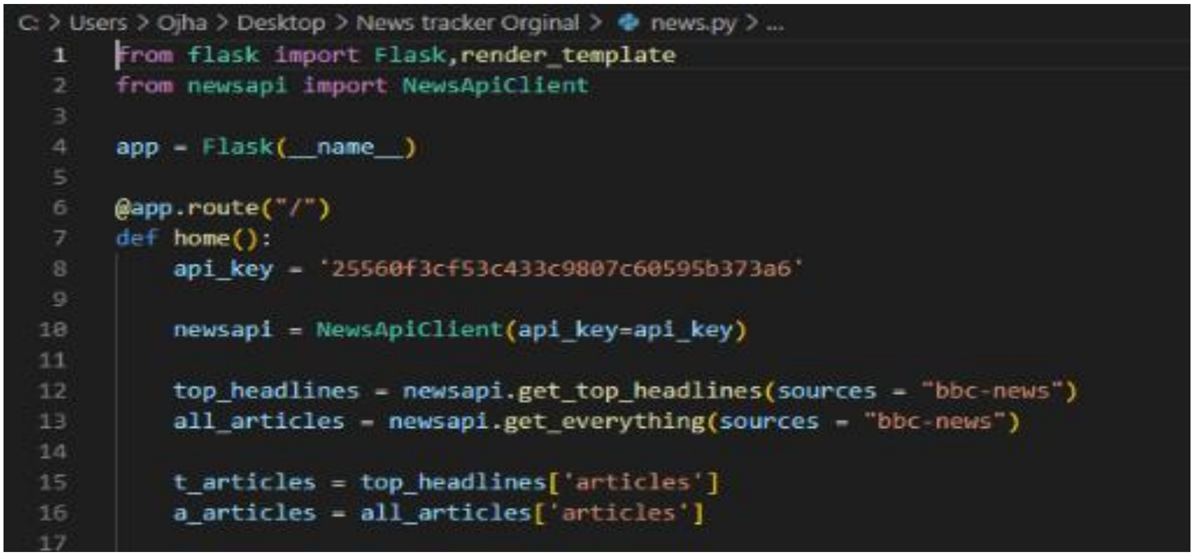
FEATURE 1 – Search Filter



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)
```

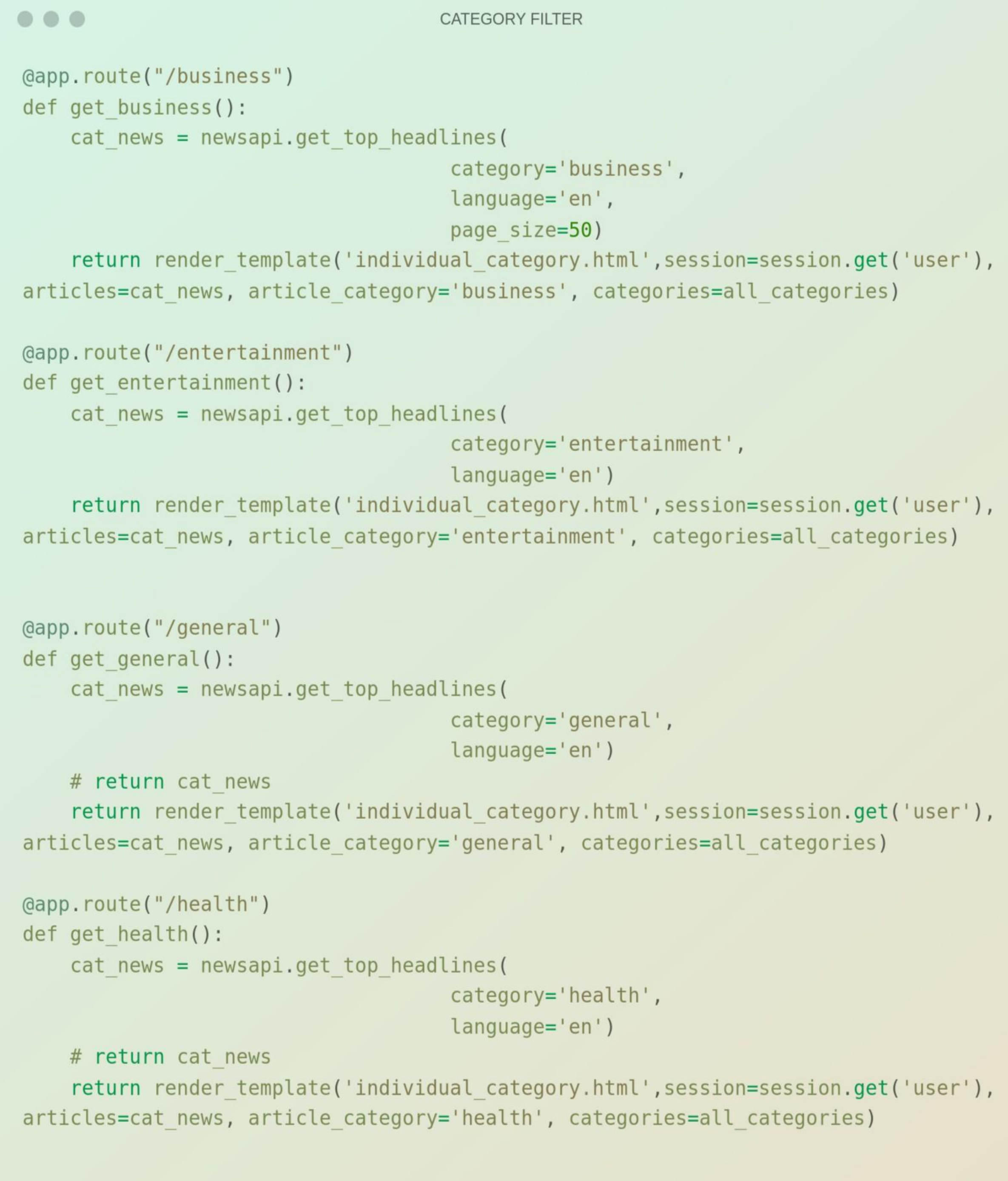
FEATURE 2



C: > Users > Ojha > Desktop > News tracker Orginal > news.py > ...

```
from flask import Flask,render_template
from newsapi import NewsApiClient

app = Flask(__name__)

@app.route("/")
def home():
    api_key = '25560f3cf53c433c9807c60595b373a6'

    newsapi = NewsApiClient(api_key=api_key)

    top_headlines = newsapi.get_top_headlines(sources = "bbc-news")
    all_articles = newsapi.get_everything(sources = "bbc-news")

    t_articles = top_headlines['articles']
    a_articles = all_articles['articles']

```

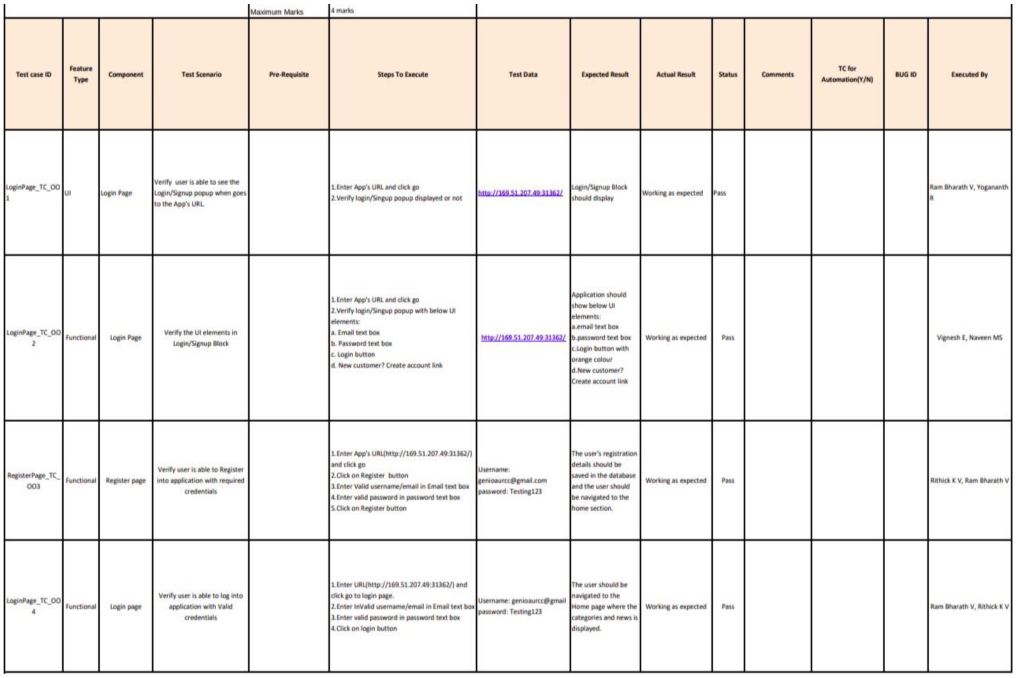
Category Filter



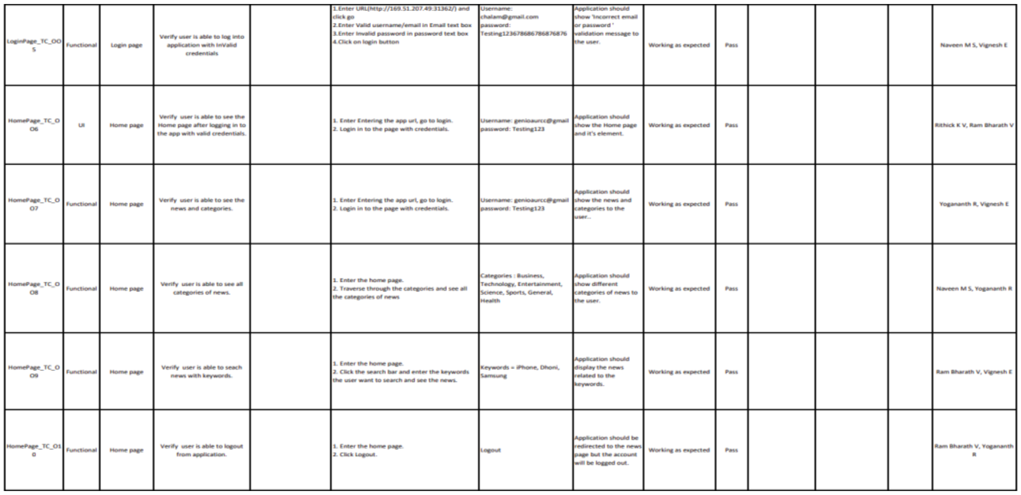
```
@app.route("/business")
def get_business():
    cat_news = newsapi.get_top_headlines(
                                    category='business',
                                    language='en',
                                    page_size=50)
    return render_template('individual_category.html',session=session.get('user'),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

Test Cases



				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	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/Singup popup displayed or not	http://169.51.207.49:31362/	Login/Signup Block should display	Working as expected	Pass				Ram Bharath V, Yogananth R
LoginPage_TC_OO2	Functional	Login Page	Verify the UI elements in Login/Signup Block		1.Enter App's URL and click go 2.Verify login/Singup popup with below UI elements: a. Email text box b. Password text box c. Login button d. New customer? Create account link	http://169.51.207.49:31362/	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, Naveen MS
RegisterPage_TC_OO3	Functional	Register page	Verify user is able to Register into application with required credentials		1.Enter App's URL(http://169.51.207.49:31362/) 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: genioaurcc@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				Rithick K V, Ram Bharath V
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Valid credentials		1.Enter URL(http://169.51.207.49:31362/) 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: genioaurcc@gmail password: Testing123	The user should be navigated to the Home page where the categories and news is displayed.	Working as expected	Pass				Ram Bharath V, Rithick K V



LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(http://169.51.207.49:31362/) 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: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message to the user.	Working as expected	Pass				Naveen M S, Vignesh E
HomePage_TC_OO6	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: genioaurcc@gmail password: Testing123	Application should show the Home page and it's element.	Working as expected	Pass				Rithick K V, Ram Bharath V
HomePage_TC_OO7	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: genioaurcc@gmail password: Testing123	Application should show the news and categories to the user..	Working as expected	Pass				Yogananth R, Vignesh E
HomePage_TC_OO8	Functional	Home page	Verify user is able to see all categories of news.		1. Enter the home page. 2. Traverse through the categories and see all the categories of news	Categories : Business, Technology, Entertainment, Science, Sports, General, Health	Application should show different categories of news to the user.	Working as expected	Pass				Naveen M S, Yogananth R
HomePage_TC_OO9	Functional	Home page	Verify user is able to seach 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, Dhoni, Samsung	Application should display the news related to the keywords.	Working as expected	Pass				Ram Bharath V, Vignesh E
HomePage_TC_O10	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 news page but the account will be logged out.	Working as expected	Pass				Ram Bharath V, Yogananth R

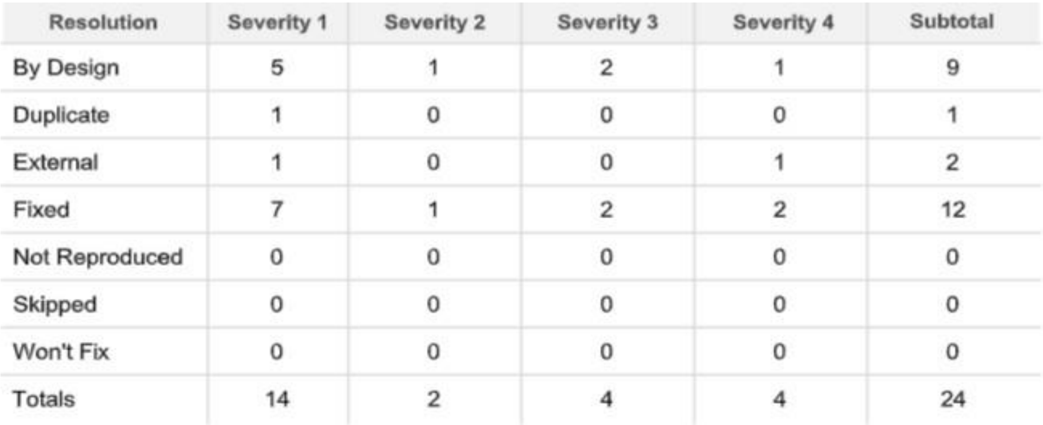
User Acceptance Testing

1.PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issuesof 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.

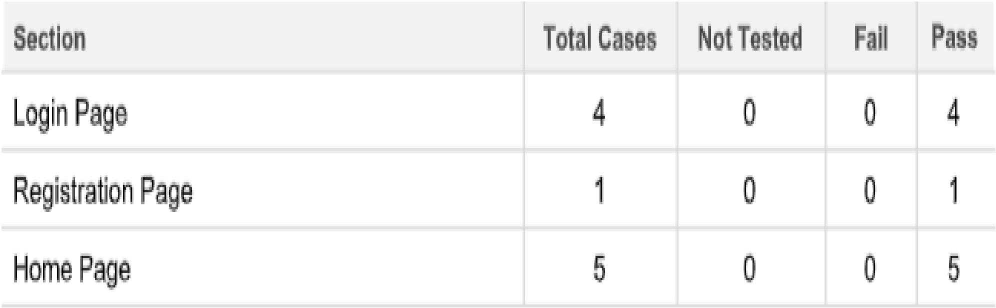


Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	1	2	1	9
Duplicate	1	0	0	0	1
External	1	0	0	1	2
Fixed	7	1	2	2	12
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	14	2	4	4	24

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 inpanic 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'treach to certain article source it gives null value which can cause problem in JSON parsing .

12. APPENDIX

SURCE 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))
```