1.INTRODUCTION:

● As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule.

● Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day.

● Only read what you feel relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Futures, Rates, Bonds, etc.… as on official websites.

1.1PROJECT OVERVIEW

The main focus of this application is to connect news articles from all around the world and deliver it to user as fast as possible in best visualize way.

1.2PURPOSE

A news application is a big interactive database that tells a news story. Think of it like you would any other piece of journalism. It just uses software instead of words and pictures.

2.LITERATURE SURVEY

2.1EXISTING PROBLEM

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 which gets shared. 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.

2.2REFERENCES

1.Title:

Exploring mobile news reading interactions for news app personalisation

Author:

Marios Constantini des, John Dowell, David Johnson, Sylvain Malacria

Date published:

July 2018

2.Title:

Topic Detection and Tracking in News Articles

Author:

Sagar Patel Nehal Patel Sandeep Patel

Date published:

March 2017

3.Title:

An Improved Method for Multi-Lingual News Feed Application

Author:

 Regonda Nagaraju, Mohammed Farhan Pasha, Mohammed Abdul Majeed, AdapaSujit h

Date published:

October 2019

4.Title:

Tracking Digitally Consumer News

Author:

Martijn Kleppe & Irene Costera Meijer

Date published:

May 2015

2.3Problem Statement Definiton

As news is increasingly accessed on smartphones and tablets, the need for personalizing news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces. We first surveyed users' news reading preferences and behaviors; analysis revealed three primary types of reader. We then implemented and deployed an Android news app that logs users' interactions with the app. We used the logs to train a classifier and showed that it is able to reliably recognise a user according to their reader type. Finally we evaluated alternative, adaptive user interfaces for each reader type. The evaluation demonstrates the differential benefit of the adaptation for different users of the news app and the feasibility of adaptive interfaces for news apps

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Says and do:

•users want reliable news

•Only read what you feel relevant and save your time

•Changing the conventional method of reading news can help

•Just tell us what market news you're interested in and get a quick peek for the day

Thing and feel:

•OMG! Spreading fake news

•I Think This News Papers Needs Almost Two Trees

•Daily News Are Coming Delay

•There is any one to tell todays new

See:

•User can read news at anytime,anywhere

• It save time for users

•This application update the news rapidly and send the updated news to the users

Here:

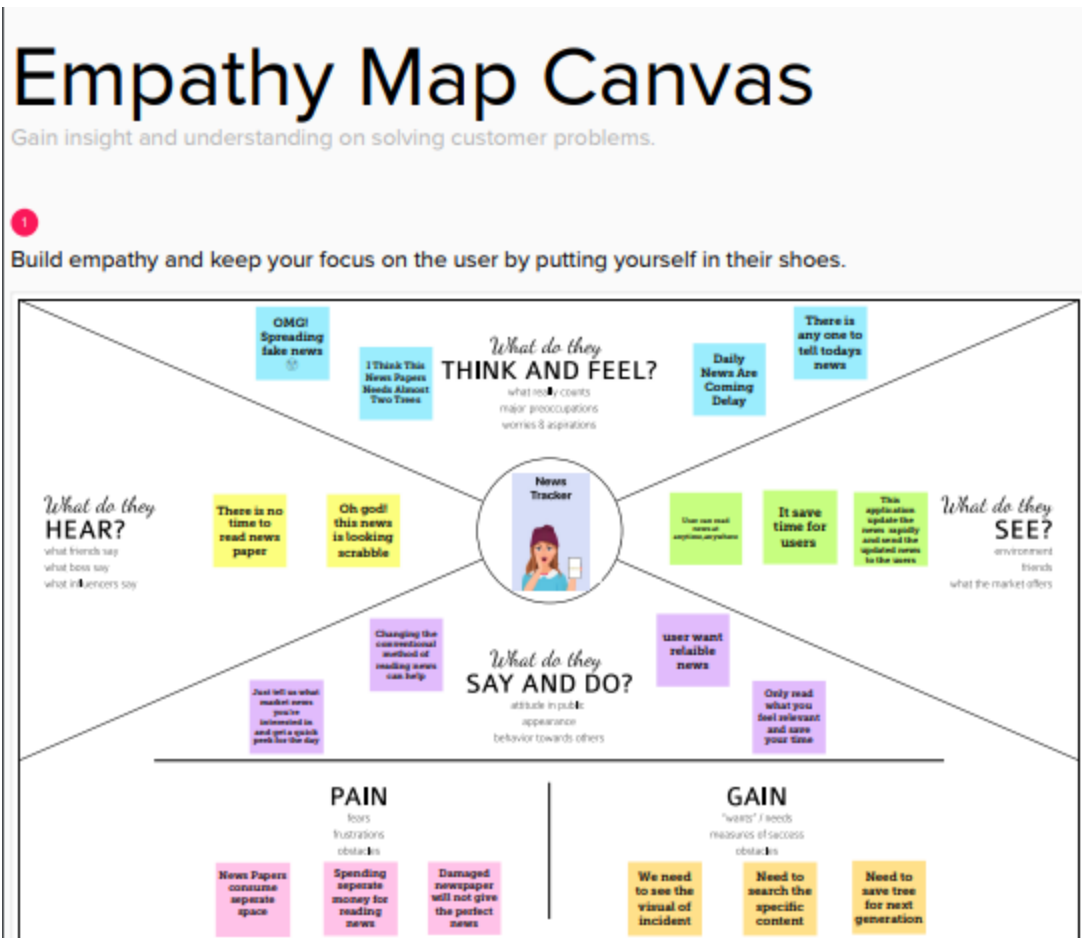•There is no time to read news paper

•Oh god! this news is looking scrabble

Pain:

•News Papers consume seperate space

•Spending seperate money for reading news

•Damaged newspaper will not give the perfect new
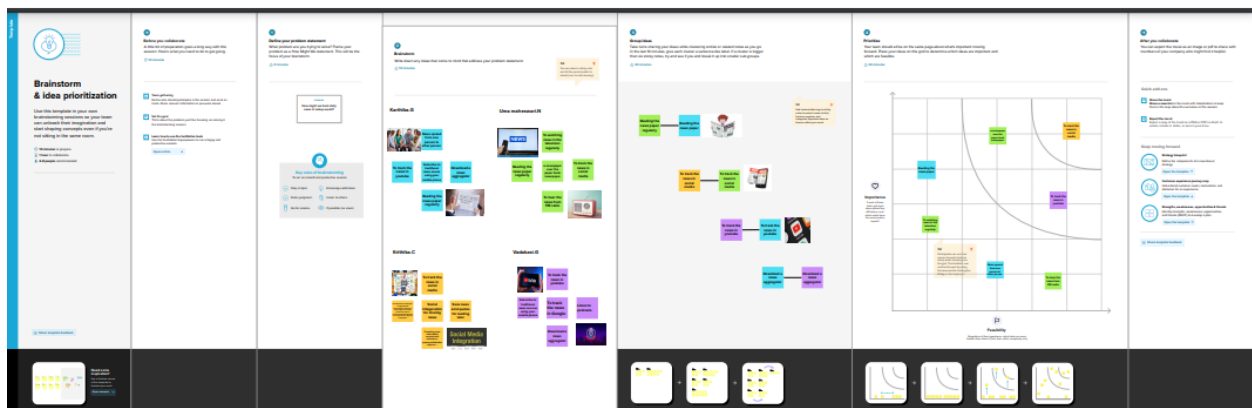
Gain:

•We need to see the visual of incident

•Need to search the specifc content

•Need to save tree for next generation

3.2 IDEATION&BRAINSTORM

3.3 PROPOSED SOLUTION

•Problem Statement (Problem to be solved):

As our lives are very busy these days,we often feel we need more than 24hrs.So we can't have separate time for reading the newspaper and watch tv channel.People need spend extra money for newspaper.

•Idea / Solution description:

This app helps you to query for all information about indices,commodities,currencies,futures,rates,bonds,etc…as on official websites.

•Novelty / Uniqueness:

News can view in any time and any where

•SocialImpact:

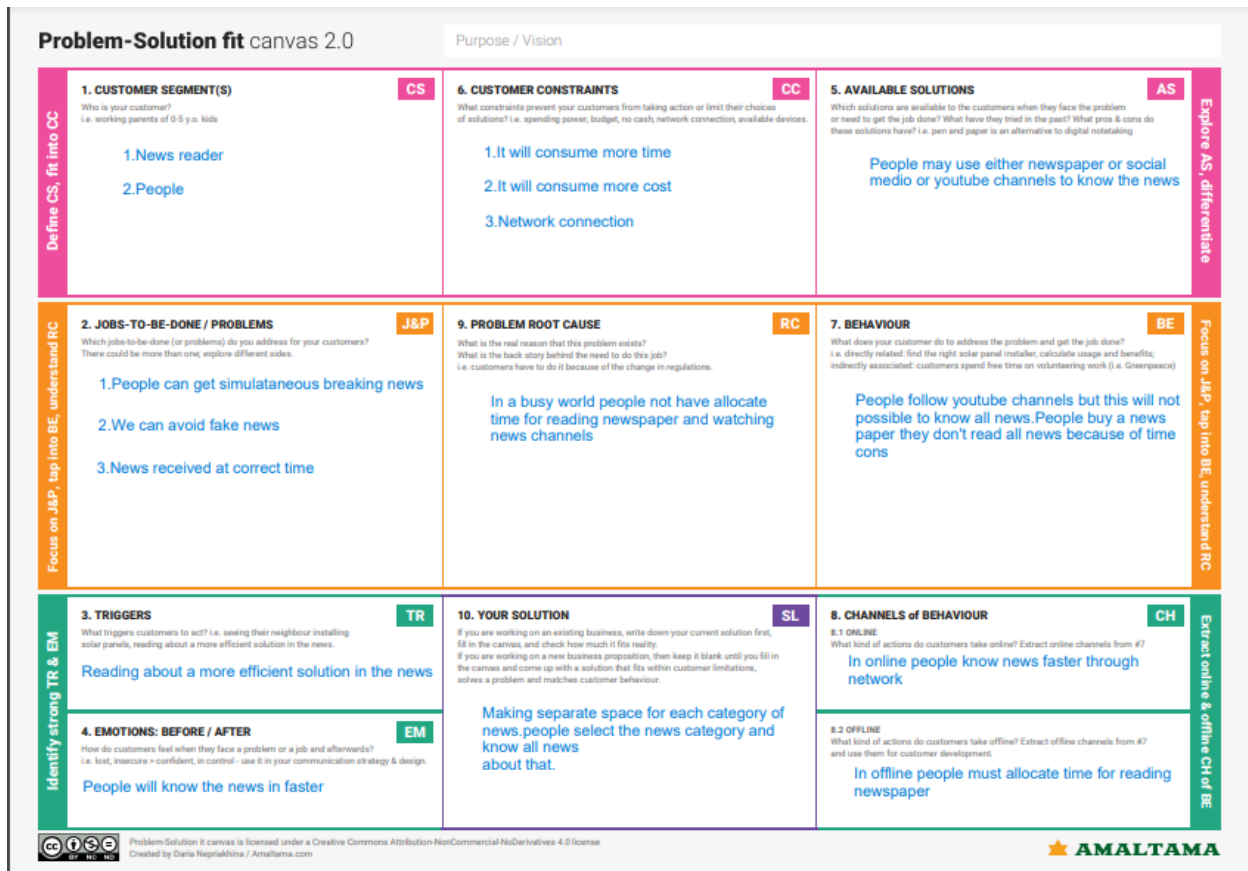 Helps news ereaders to get required information within less tim.

•Business Model:

We can include the adds in portal.

•ScalabilityoftheSolution:

Performance and speeddonotslowdowneven if largenumberofusersaccesstheapplication atthesametime.

3.4 PROBLEM SOLUTION FIT

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposedsolution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration throughForm
FR-2	User Confirmation	Confirmationvia Email
FR-3	User Login	Login Via EmailID & Password
FR-4	Home Page	Shows the NEWSHeadlines.
FR-5	Tracker Page	Shows the NEWSelaboratively when the NEWS Headline of the Home pageis clicked.

4.2NON FUNCTIONAL REQUIREMENT

Following are the non-functional requirements of the proposedsolution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Our Website is used to Read Newsand gain Knowledge according to user's interest.
NFR-2	Security	User can only login with their own Email ID and password withouttheir Emailand password no one canuse the App.
NFR-3	Reliability	Our Website delivers the reliable content only from NEWSAPI.
NFR-4	Performance	It is a high performable App with friendly User Interface and shares lot of information without any Issues.
NFR-5	Availability	Our Website is supported withall search Engineso it willAvail fo Userat any time.

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.
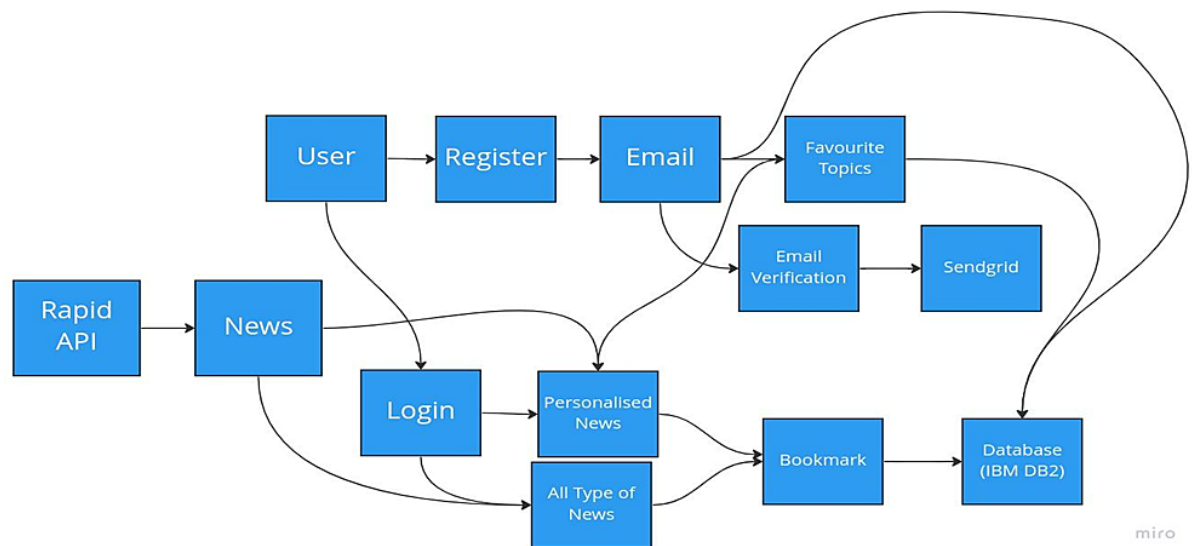
5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture:
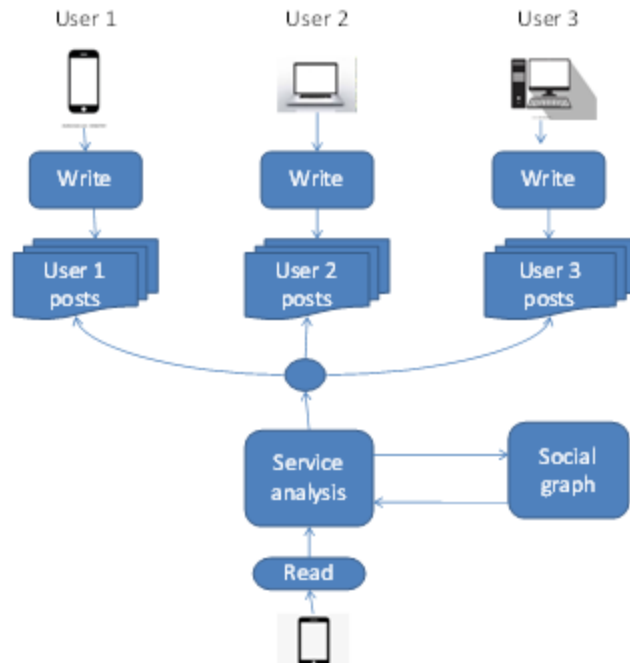
Technical Architecture:
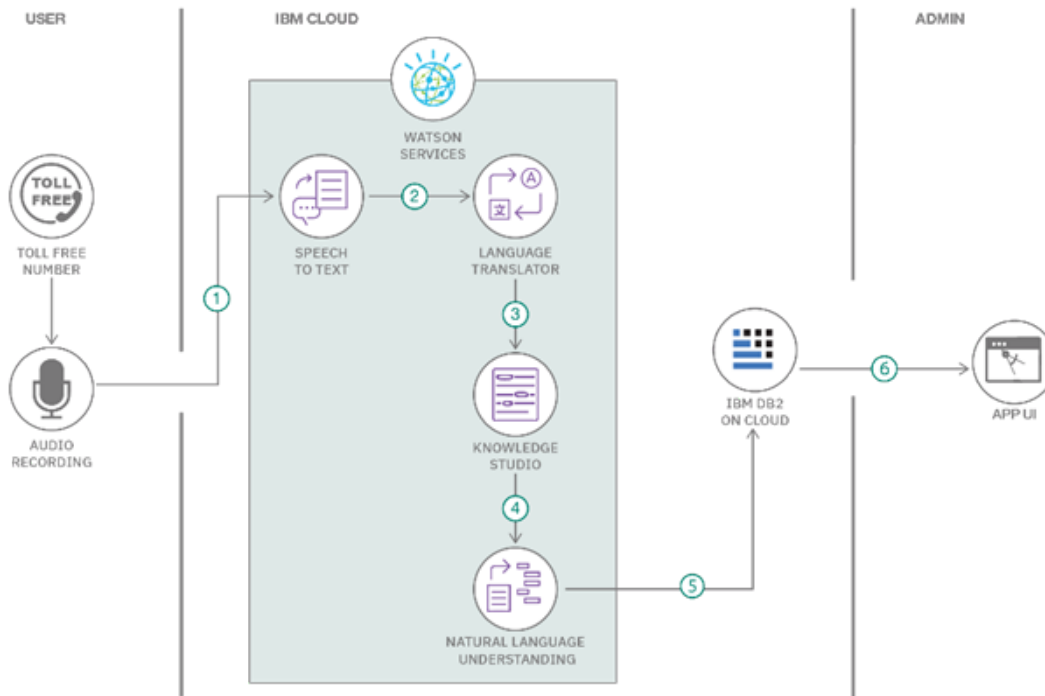
5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance crit
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 / dashb
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation em & click confirm

		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login
		USN-4	As a user, I can register for the application through Gmail	
	Login	USN-5	As a user, I can log into the application by entering email & password	
	Dashboard			
Customer (Web user)				
Customer Care Executive				
Administrator				

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Project Planning Phase

NEWS TRACKER APPLICATION

Project Planning Template (ProductBacklog Sprint Planning, Stories,Storypoints)

Date	16 nov 2022
Team ID	PND2002TMID50005
Project Name	News TrackerApplication
Maximum mark	8mark

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create productbacklog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority
Sprint-1	User Login	USN-1	As a user, I can log into the application by entering email & password	MEDIUM
	User Registration	USN-2	As a user, I can register for the application by entering my email, password and confirming my passwordand phonenumber.	
Sprint-2	User searchpage	USN-3	Even without logging in User must be able to search for news in the home page	HIGH

		USN-4	After logging in, news specified in his favorite category must be grouped in the top	Low

		USN-5	I must be ableto visit the source news website	Low
Sprint-3				
	User NEWS selection choice	USN-6	As a user, I must be able to choose categories based upon my interest	HIGH
Sprint-4	User notification	USN-7	As a user, I must be notified with my favorite news through mail	HIGH
		USN-8	As a user I mustbe able to specifywherer I must be notified	low

Project tracker, velocity&burn down chart:(4 marks)

sprint	Total story points	duration	Sprint start date	Sprint end date(planned)	Story completed (as on planned end date)	Sprint release date(actual)

Sprint -1	1	3days		24 oct 2022	26 oct 2002	1
Sprint -2	1	3days		31 oct 2002	02 nov 2022	1
Sprint -3	1	3days		7 nov 2022	09 nov 2022	1
Sprint -4	1	3days		14 nov 2022	16 nov 2022	1

Velocity:

Imagine we have 10-days sprint duration and the velocity of the team is 20 (points per sprint).let's calculate the team's average velocity (AV) per iteration unit (storypoint per day)

$$AV = \text{sprint duration/ velocity}$$

$$= 20/10$$

$$= 2$$

6.2 SPRINT DELIVERY SCHEDULE

Project Planning Phase

Project Planning Template (ProductBacklog Sprint Planning, Stories,Storypoints)

Date	16 nov 2022
Team ID	PND2002TMID50005
Project Name	News TrackerApplication
Maximum mark	8mark

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create productbacklog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority
Sprint-1	User Login	USN-1	As a user, I can log into the application by entering email & password	MEDIUM

	User Registration	USN-2	As a user, I can register for the application by entering my email, password and confirming my passwordand phonenumber.	
Sprint-2	User searchpage	USN-3	Even without logging in User must be able to search for news in the home page	HIGH
		USN-4	After logging in, news specified in his favorite category must be grouped in the top	Low

Sprint-3		USN-5	I must be ableto visit the source news website	Low
	User NEWS selection choice	USN-6	As a user, I must be able to choose categories based	HIGH

| Sprint-4 | User notification | USN-7 | As a user, I must be notified with my favorite news through mail | HIGH |
| | | USN-8 | As a user I mustbe able to specifywherer I must be notified | low |

Project tracker, velocity&burn down chart:(4 marks)

sprint	Total story points	duration	Sprint start date	Sprint end date(planned)	Story completed (as on planned end date)	Sprint release date(actual)
Sprint -1	1	3days		24 oct 2022	26 oct 2002	1
Sprint -2	1	3days		31 oct 2002	02 nov 2022	1
Sprint	1	3days		7 nov 2022	09 nov	1

-3					2022	
Sprint -4	1	3days		14 nov 2022	16 nov 2022	1

Velocity:

Imagine we have 10-days sprint duration and the velocity of the team is 20 (points per sprint).let's calculate the team's average velocity (AV) per iteration unit (storypoint per day)
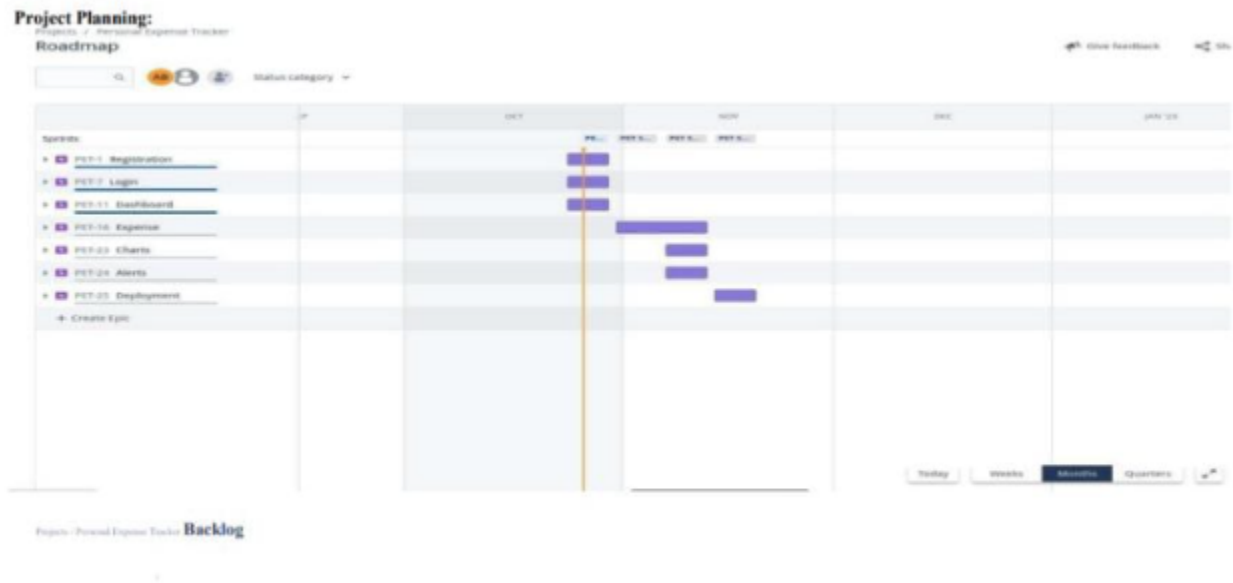
$$AV = sprint\ duration/\ velocity$$

$$= 20/10$$

$$= 2$$

6.3 REPORT FROM JIRA

7.CODING AND SOLUTIONING

7.1 FEATURE 1

•categories
•search and filters
•push notification
•engagement tools

7.2 FEATURE 2

•social media integration
•registration and personal profiles
•personal feed
•options to interact with content

8.TESTING

8.1 TEST CASES

•Check if the app is able to start and stop and behaves as per the requirement
Behavior of the app in the background.
•The behavior of the app at the time of interruptions such as phone calls, notifications, etc.
Scroll functionality, to check the smoothness of the page while scrolling.
•Check whether the auto-logout functionality is working as per the requirement.

8.2 USER ACCEPTANCE TESTING

GOAL: The main goal of this application is to connect news articles from all around the world
and deliver it to user as fast as possible in best visualize way.

SCOPE:

A news app needs credibility and a name to which it is associated and if not exactly there will be some apps
that revolve around the same idea.

9.RESULTS

9.1 PERFORMANCE METRICS

•Acquisition. Downloads: Number of new downloads in a given timeframe (daily, weekly, or monthly)

•Activations. Activation Rate: Percentage of downloads that launched the app

•Retention. Simply put, retention = revenue

•Engagement

•Drop Offs

•Reachability.

10.ADVANTAGES:

- Own Your Channels. ...
- Better User Experience. ...
- Higher Engagement. ...
- Push notifications. ...
- Revenue Opportunities. ...
- App Store Presence.

DISADVANTAGES:

- Require data/wifi to get online.
- Companies not making as much money due to free reading for audiences.
- News spreads quicker online - people find out news before they should.
- Lose money - can't get people to pay for digital.
- Older audiences may not access digital platforms.
- Costly to maintain.

11.CONCLUSION

Online newspapers provide a challenge for the theories we employ to analyse and understand texts, for the critical analysis of such texts, and for the development of pedagogical approaches by which knowledge and understanding of such texts can be taught and learned. These challenges are significant, and meeting them will require a range of approaches, and sustained effort. The current research has implications for meeting these challenges, and is intended as a contribution towards understanding of online newspapers and the roles they play in a range of social contexts.

12.FUTURE SCOPE

A news app has simple goals: Capture and retain reader attention, and repeat the process, several times a day. Pretty straightforward. But not that simple in the real world. For a news provider, the smartphone screen is the the most challenging environment ever seen. There, chances are that a legacy media or a pure-player will find itself in direct competition, not only with the usual players in its field, but also with Facebook, Snapchat, Instagram and scores of gaming applications. Distraction is just one icon away; any weakness in functional or graphic design can be lethal.

13.APPENDIX

13.1.SOURCE CODE

home.html

```html
<html>


<html lang="en">

<head>

<title>News Tracker</title>

<link rel="stylesheet" type="text/css" href="{{url_for('static',filename='newstyle.css')}}">

<link rel="stylesheet" type="text/css" href="{{url_for('static',filename='font/css/all.min.css')}}">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css"/>

</head>

<body>

<div class="News">

<header>

<div class="logo">

<pre style="font-size:20px ; font-family:Times New Romon; ">News Tracker

</pre>

</div>

<div class="search-box">

<i class="fa-solid fa-magnifying-glass"></i>

<input type="search" placeholder="Search">

</div>

<nav>

<ul>

<li><a href="/Home"><i class="fa-regular fa-address-card"></i>  Home</a></li>

<li><a href="/Local"><i class="fa-regular fa-address-card"></i>  Local</a></li>

<li><a href="/Coronavirus"><i class="fa-regular fa-address-card"></i>  Coronavirus</a></li>

<li><a href="/about1" ><i class="fa fa-edit"></i>aboutus </a></li>

<li><a href="/register1" ><i class="fa fa-edit"></i> Register</a></li>
```

```
<li><a href="/login1" ><i class="fa fa-edit"></i> login</a></li>

<ul>

</nav>

</header>

<section>

<div class="posts">

<div class="post-title">

<div class="post-left">

<div class="image">

</div>

<center>

<div>

<div class="details">

<p class="name">The Hindu</p>

<p class="location">Tamil Nadu</p>

</div></center>

<div class="post-right">

<i class="fa-solid fa-ellipsis"></i>

</div>

<div class="post-content">
```

```
<center><img src="{{url_for('static',filename='image/news2.webp')}}" style="width:600; height:600;"></center>

</div>

<div class="post-footer">

<div class="like-share-comment">

<i class="fa-regular fa-heart"></i>

<i class="fa-regular fa-comment"></i>

<i class="fa-solid fa-share"></i>

</div>

<div class="save">

<i class="fa-solid fa-bookmark"></i>

</div>

<div class="post-footer-content">

<p class="likes">100likes</p>

<p class="comments">view all 10 comments</p>

<p class="posting-time">23 hours ago</p>

</div>

<div class="add-comment">

<div class="left-side">

<i class="fa-regular fa-face-smile"></i>

<input type="text" placeholder="Add a commant.....">
```

```
</div>




</body>

</html>
```

app.py
```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)


Conn=ibm_db.connect("database=bludb;hostname=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;port=30875;SECURITY=SSL;SSLServerCe
rtificate=DigiCertGlobalRootCA.crt;uid=brv13326;pwd=c3GnX6SgfSkf3dyS" , ",")

@app.route('/')
def home():
    return render_template('home.html')
@app.route('/login1')
def login1():
    return render_template('login.html')
@app.route('/register1')
def register1():
    return render_template('register.html')
@app.route('/about1')
def about1():
    return render_template('about.html')
@app.route('/help1')
```

```python
def help1():
    return render_template('help.html')
@app.route('/report')
def report():
    return render_template('report.html')


@app.route('/login', methods =['GET', 'POST'])
def login():
    global userid
    msg = ''
    if request.method == 'POST'and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        stmt = ibm_db.prepare(Conn,'SELECT * FROM users WHERE username = ? AND password = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('home2.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))
@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
```

```python
        phone_no=request.form['no']
        sql = "SELECT * FROM users WHERE username = ? "
        stmt = ibm_db.prepare(Conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?,?)"
            stmt = ibm_db.prepare(Conn,insert_sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.bind_param(stmt, 4, phone_no)
            ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'

    return render_template('register.html', msg = msg)
if __name__=='__main__':
    app.run(debug=True)
```

business.html
 extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
   <div class="row">

```
{% for source, title, desc, author, img, p_date, url in sources %}
<div class="col-md-4 d-flex justify-content-center">
  <div class="card mb-3 bg-dark text-white">
    <img src="{{ img }}" class="card-img image-fluid" alt="...">
    <div class="card-body">
      <p class="card-text source">{{ source.name }}</p>
      <a href="{{ url }}">
        <h2 class="card-title"><b> {{ title }} </b></h2>
      </a>
      <p class="card-text">{{ desc }}</p>
      <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
    </div>
    <div class="card-footer">
      <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p_date }}</p>
    </div>
  </div>
</div>
{% endfor %}
</div>
</div>

{% endblock %}


{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
  <div class="row">
    {% for source, title, desc, author, img, p_date, url in sources %}
    <div class="col-md-4 d-flex justify-content-center">
```

```
        <div class="card mb-3 bg-dark text-white">
            <img src="{{ img }}" class="card-img image-fluid" alt="...">
            <div class="card-body">
                <p class="card-text source">{{ source.name }}</p>
                <a href="{{ url }}">
                    <h2 class="card-title"><b> {{ title }} </b></h2>
                </a>
                <p class="card-text">{{ desc }}</p>
                <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0
0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579
10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3
3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12
19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
            </div>
            <div class="card-footer">
                <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 5c-
4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6 2.691 6 6-
2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2
2z"></path></svg> Date Published: {{ p_date }}</p>
            </div>
        </div>
    {% endfor %}
  </div>
</div>

{% endblock %}




headline.html
{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
    <div class="row">
        {% for source, title, desc, author, img, p_date, url in headlines %}
        <div class="col-md-4 d-flex justify-content-center">
```

```
        <div class="card mb-3 bg-dark text-white">
            <img src="{{ img }}" class="card-img image-fluid" alt="...">
            <div class="card-body">
                <p class="card-text source">{{ source.name }}</p>
                <a href="{{ url }}">
                    <h2 class="card-title"><b> {{ title }} </b></h2>
                </a>
                <p class="card-text">{{ desc }}</p>
                <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0
0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579
10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3
3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12
19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
            </div>
            <div class="card-footer">
                <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 5c-
4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6 2.691 6 6-
2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2
2z"></path></svg> Date Published: {{ p_date }}</p>
            </div>
        </div>
    </div>
    {% endfor %}
  </div>
</div>

{% endblock %}

 sports.html
{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
  <div class="row">
    {% for source, title, desc, author, img, p_date, url in sources %}
    <div class="col-md-4 d-flex justify-content-center">
      <div class="card mb-3 bg-dark text-white">
        <img src="{{ img }}" class="card-img image-fluid" alt="...">
```

```
        <div class="card-body">
          <p class="card-text source">{{ source.name }}</p>
          <a href="{{ url }}">
            <h2 class="card-title"><b> {{ title }} </b></h2>
          </a>
          <p class="card-text">{{ desc }}</p>
          <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
        </div>
        <div class="card-footer">
          <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p_date }}</p>
        </div>
      </div>
    </div>
    {% endfor %}
  </div>
</div>

{% endblock %}

tech.html
{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
  <div class="row">
    {% for source, title, desc, author, img, p_date, url in sources %}
    <div class="col-md-4 d-flex justify-content-center">
      <div class="card mb-3 bg-dark text-white">
        <img src="{{ img }}" class="card-img image-fluid" alt="...">
        <div class="card-body">
          <p class="card-text source">{{ source.name }}</p>
```

```html
                <a href="{{ url }}">
                  <h2 class="card-title"><b> {{ title }} </b></h2>
                </a>
                <p class="card-text">{{ desc }}</p>
                <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
              </div>
              <div class="card-footer">
                <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;"><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p_date }}</p>
              </div>
            </div>
          </div>
          {% endfor %}
        </div>
      </div>

{% endblock %}

style.css
* {
   font-family:'Overpass', Tahoma, Geneva, Verdana, sans-serif;
   background-color: #1e1e21;
   color: #fff;
}

p{
   font-size: 14px;
}

a {
   text-decoration: none;
   color: #fff;
}
```

```css
a:hover {
   text-decoration: none;
   color: #fff;
}

.landing {
   margin-top: 100px;
}

#navbarToggle {
   background-color: #000;
}

.navbar.navbar-dark .nav-item .nav-link {
   color: #fff!important;
   background-color: #000;
}

.navbar.navbar-dark .nav-item .nav-link:hover {
   color: #fff!important;
   background-color: #000;
}

.navbar {
   background-color: #000!important;
}

.navbar-nav {
   background-color: #000;
}

.navbar-brand {
   background-color: #000;
}

.lg {
   background-color: #000;
   font-size: xx-large;
}
```

```css
h1 {
   font-size: 33px;
   color: #dfdcd9;
}

h2 {
   font-size: 1.5em;
   color: #dfdcd9;
}

.author {
   font-size: 12px;
}

.date {
   font-size: 12px;
}

.article-card {
   position: relative;
   display: -ms-flexbox;
   display: flex;
   -ms-flex-direction: column;
   flex-direction: column;
   min-width: 0;
   color: #000;
   background-color: #fff;
   word-wrap: break-word;
   background-clip: border-box;
   border-radius: 10px;
   margin-top: 15px;
}

.img {
   width: 100%;
}

.image-fluid {
   height: 18em;
```

```css
}

.cardfooter {
    padding: 0.75rem 1.25rem;
    padding-bottom: 0px;
    border-radius: 0px 0px 10px 10px;
}

.article_name {
    color: #dfdcd9;
}

.source-body {
    margin-bottom: 30px;
}

.source {
    margin-top: 5px;
    padding-top: 5px;
    padding-bottom: 0px;
    margin-bottom: 0px;
    color: #dd0000;
}

.logo {
    color: #dd0000;
    background-color: #000;
}

.svg {
    background-color: #000;
}

.p-svg {
    background-color: #26262a;
}

.footer {
    background-color: #000;
    padding-top: 20px;
```

```css
    padding-bottom: 15px;
}

.footer-text {
    background-color: #000;
    margin-bottom: 0px;
}

.dropdown-menu {
    background-color: #1e1e21;
}

.dropdown-item {
    color: #fff;
}

.dropdown-item:hover {
    color: #fff;
    background-color: #26262a;
}

.published {
    background-color: #26262a;
}

.card-footer {
    background-color: #26262a;
}

newstyle.css
*{
margin:0;
padding:0;
box-sizing:border-box;
font-family:sans-serif;
}
/*header*/
header{
display:flex;
justify-content:space-between;
```

```css
    align-items:center;
    padding:10px 18%;
    border-bottom:1px solid rgba(219,219,219,1);
    background-color:#42eff5 ;


}
header .logo{
    height:29px;
    float:left;

}
header .logo pre{
    height:100%;
    object-fit:contain;
}
header .search-box{
    width:250px;
    background-color:rgba(239,239,239,1);
    padding:3px 16px;
    display:flex;
    align-items:center;
    border-radius;30px;
}
header .search-box input{
    height:30px;
    border:none;
    outline:none;
    background-color:rgba(239,239,239,1);
    padding-left:10px;
}
header nav ul{
    list-style:none;
    display:flex;
}
header nav ul li{
    margin-right:20px;
}
/*posts*/
.posts{
```

```
margin:25px 0;
background-color:white;
border:1px solid rgba(219,219,219,1);
border-radius:3px;
}
. posts .posts-title{
display:flex;
justify-content:space-between;
align-items:center;
padding:16px;

}
```

13.2.GitHub&Demo Link

GitHub: https://github.com/IBM-EPBL/IBM-Project-41354-1660641538
Demolink:https://drive.google.com/file/d/1xpy21HawsgPn2N0-qmkVpGvmS-dHfQrS/view?usp=drivesdk

NEWS TRACKER APPLICATION