Project Report

TEAM ID : PNT2022TMID46366
PROJECT : News Tracker Application

1. INTRODUCTION

1.1. Project Overview

NewsTracker is a fullstack web application which allows users to register along withtheir favourite topics, upon login the app displays the news based on the user's interest. The news displayed in the app is based on the Newcatcher API and CricbuzzAPI from Rapid API site.A news-sharing app wants to help users find relevant and important news easily every day and also provide explicitly news from that users locality/region which may of help to the user.

1.2. Purpose

Enabling users to view news from anywhere at anytime. It also helps to reduce the timeto get information about a specific topic. Also enables a person to get an updated newswhich may help Business people to make business related decisions quickly and correctly.

2. LITERATURE SURVEY

2.1. Existing problem

Physical newspapers are old fashioned in this digital era. They cost money to buy, can easily be damaged, limited amount of information, not flexible to modifications, poor quality. Sometimes may show unrelevant and updated news.

2.2 .References

AUTHOR	PAPER TITLE	YEAR	JOURNAL	CRITICS
Martijn Kleppe and Marco Otte	Analysing and understanding news consumption patterns by tracking online user behaviour with a multimodal research Design	2017	Digital Scholarship in the Humanities	The data collection, pre-processing, and pattern discovery takes more time.
Scott R. Baker Nicholas Bloom Steven J. Davis Kyle J. Kost	Policy News and Stock Market Volatility	2019	National Bureau of Economic Research	The history of thought in financial markets has shown a surprising lack of consensus about a very fundamental question: what ultimately causes all those fluctuations in the price of speculative assets like corporate stocks.
Marios Constantinides , John Dowell , David Johnson , Sylvain Malacria	Exploring mobile news reading interactions for news app personalisation	2015	University College London	Reviewers reported they did not find the adaptive menu beneficial and would prefer a snapshot of articles within multiple categories as opposed to being restricted to one. None reported wanting article summaries despite reading long articles and being probed in the post interviews.

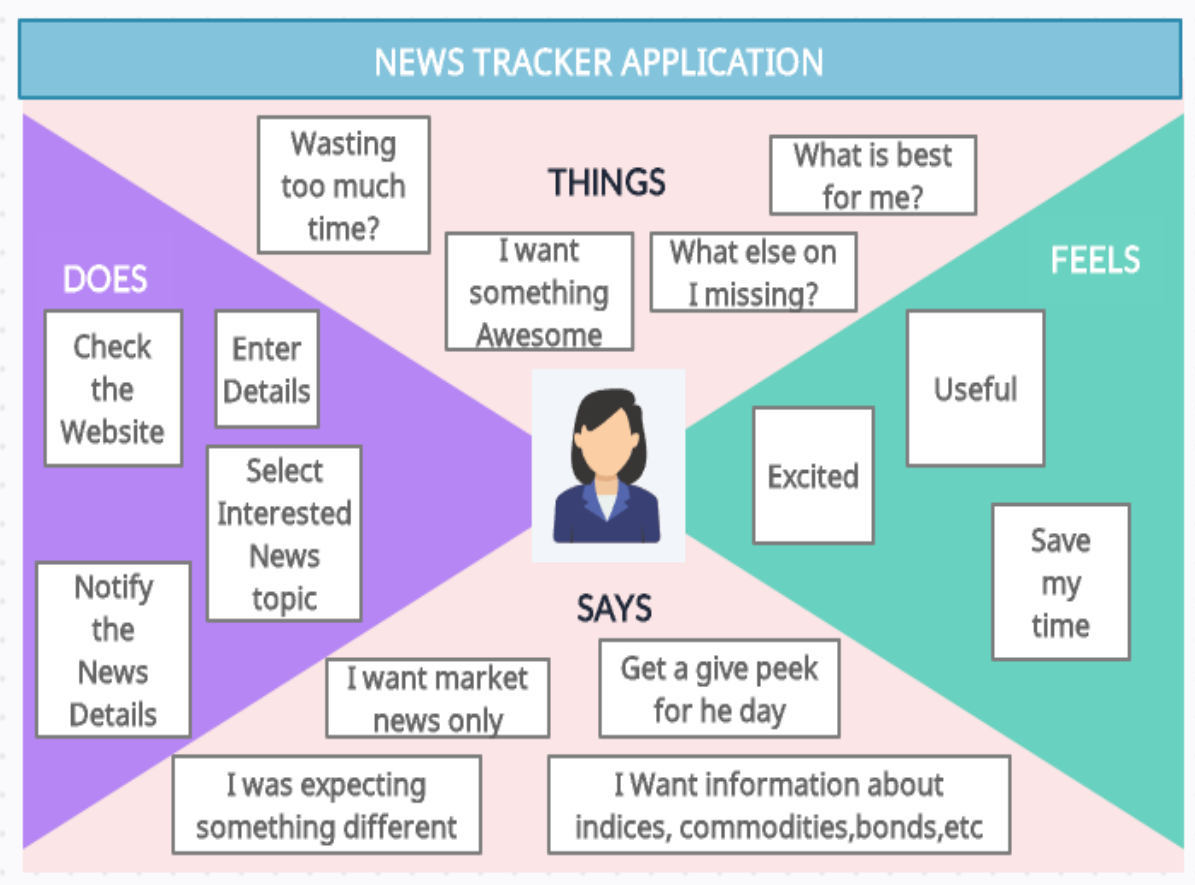
Oscar Westlund	MOBILE NEWS A review and model of journalism in an age of mobile media	2012	Digital Journalism	The importance of trust has been highlighted future research studies can focus on the antecedents of trust Users require lower level of trust.
Wei Guo and Bo Zhang	Research on Development Strategy of News App under the Background of Artificial Intelligence	2019	International Conference on AI and Big Data Application (AIBDA 2019)	Algorithm-based recommendation of this cutting-edge technology will face numerous tests from journalists, social ethics, laws and regulations.

2.3. Problem Statement Definition

Newspaper contains limited , non user/reader specific, Location specific news .There are multiple news-sharing apps available which can be used by a single user and are often spammed with notifications. There is also a lot of unwanted news which gets shared. So it may take a lot of time for the user to find the news he/she likes. A news-sharing app wants to help users find relevant and important news easily every day and also provide explicitly newsfrom that users locality/region which may of help to the user.

3. IDEATION & PROPOSED SOLUTION

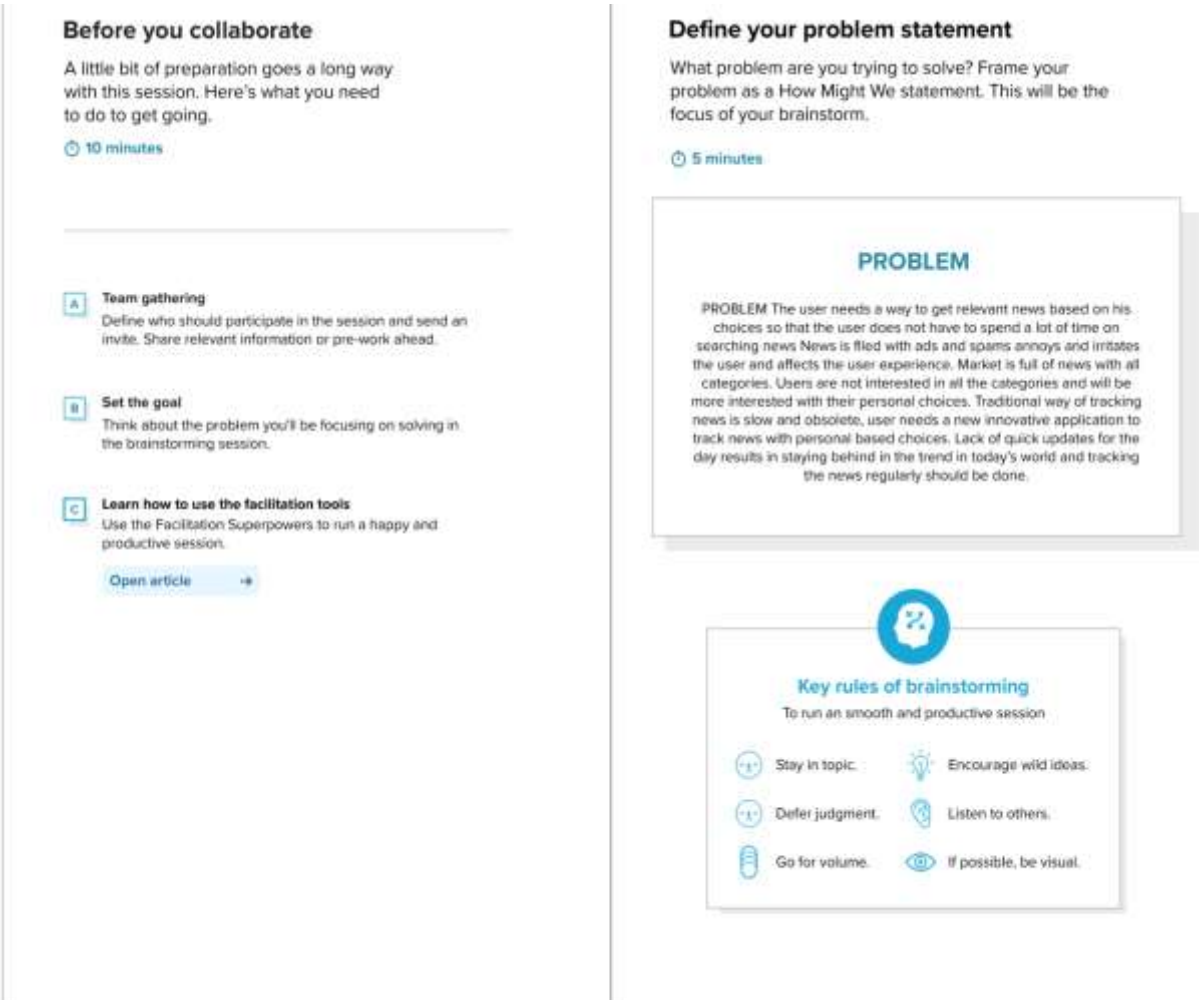
3.1.Empathy Map Canvas



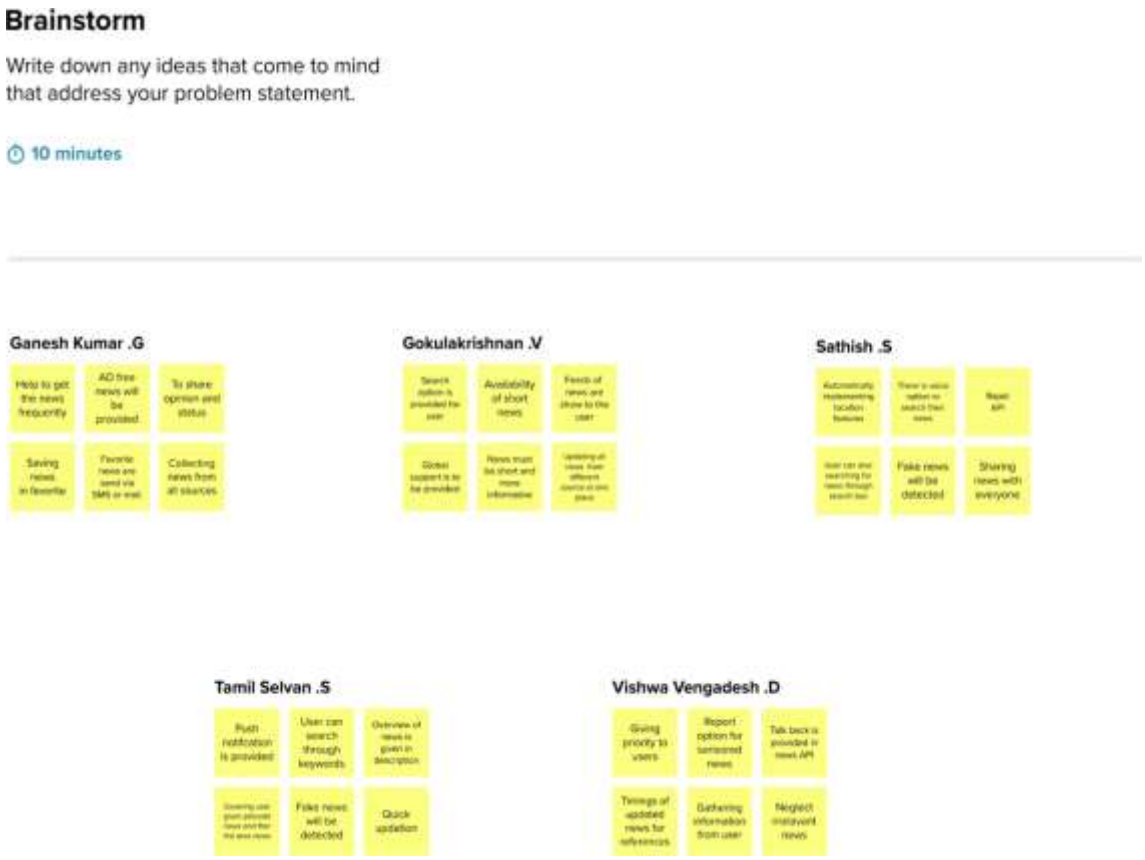
3.2. Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizingvolume over value, out-of-the-box ideas are welcome and built upon, and all participants areencouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

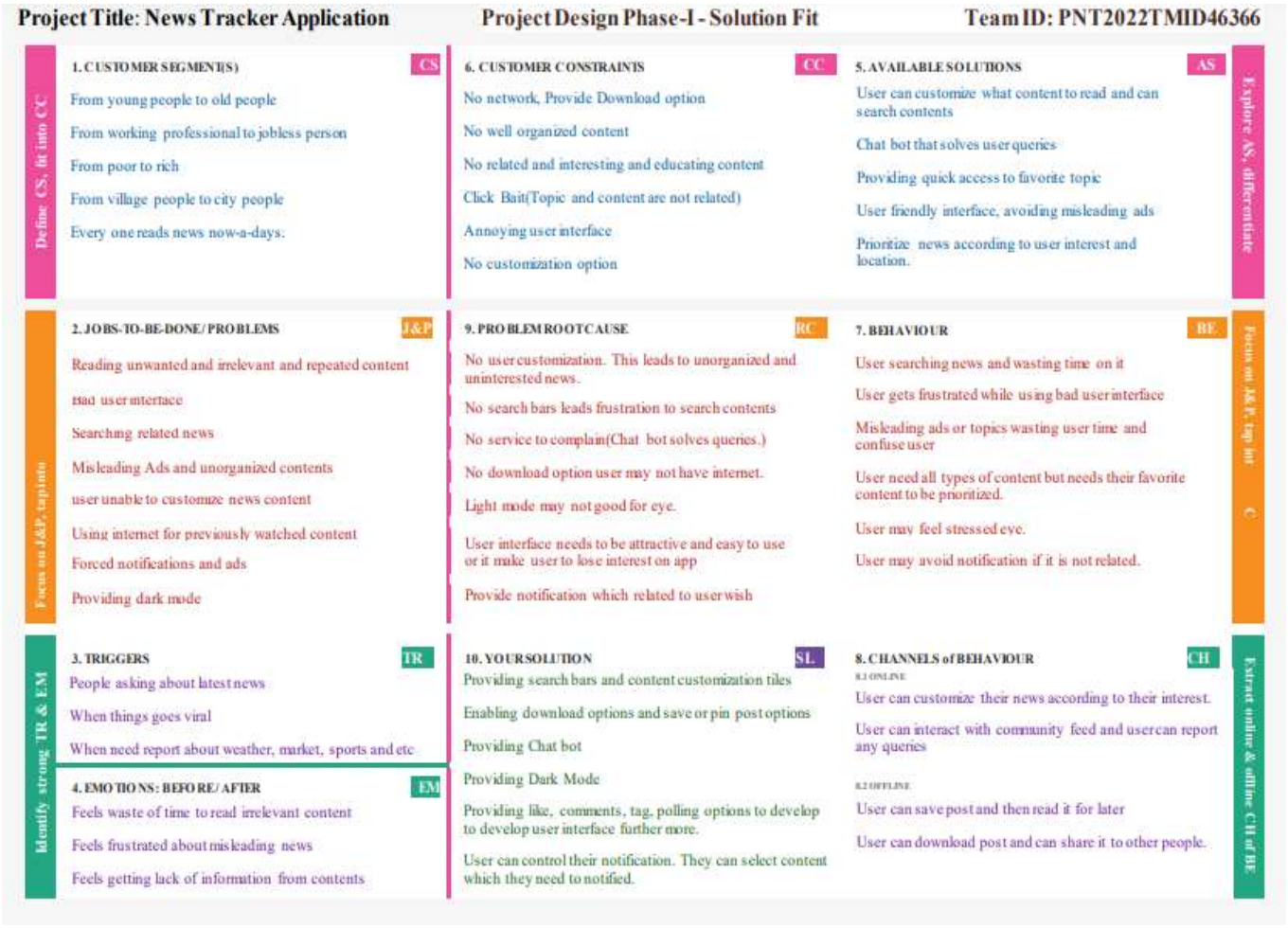


Step-3: Idea Prioritization

3.3. Proposed Solution

S. No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Most people don't like to carry a newspaper with them. Some people want them to be updated onlyin the area they are interested in.
2.	**Idea / Solution description**	An application needs to be developed in which users can read news whenever they want and they will be able to customize theirarea of interest. So that they will be notified, if any new news is updated in their interestedareas.
3.	**Novelty / Uniqueness**	1. A user can read news only from their interested fields rather than reading all the news. 2.This application provides users with a trusted and secured ecosystem. News shared through the application is original and spam free.
4.	**Social Impact / Customer Satisfaction**	Customer satisfaction can be fulfilled by this application because, it is user friendly, easily carriable, handy, etc.
5.	**Business Model (Revenue Model)**	1. Users can able to install the application from play store at a free of cost so that the number of users will get rapidly increased. 2. Users can able to install the application from play store at a free of cost so that the number of users will get rapidly increased. 3.Add premium subscription, users who subscribe for premium won't get advertisements. 4.An Advertisement-Free feature can be enabled for the users who can read or viewnews without any popup advertisements.
6.	**Scalability of the Solution**	As it was an application-based project, correct ideation and execution can develop an application with no bugs and errors, so that theuser might like our application and share it to their surroundings, resulting in an increase in our application insights.

3.4. Problem Solution Fit



.4. REQUIREMENT ANALYSIS

4.1. Functional Requirements:

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 Form ➢ Registration through Gmail
FR-2	**User Confirmation**	➢ Confirmation via Email ➢ Confirmation via OTP
FR-3	**Searching**	➢ Search the news based on the user interest ➢ Showing the trending news on search ➢ Showing the Category- wise news ➢ User can save the news needed and study later ➢ Help users find content with categorized
FR-4	**Real time News**	➢ User can see the Real news with Real time update ➢ Auto update news because it fetch news from API ➢ Show the Number of comments an article as a and the number of comment can be used as a measure of comment popularity
FR-5	**Location based**	➢ User can view the news near to their location ➢ User can track the Location of the news
FR-6	**User friendly**	➢ User can publish their own articles ➢ User can add images to their articles ➢ Users can see the articles Published by the other users ➢ Push notification are meant to attract users attention by using alerts, even if it's just for a second.

4.2. Non-functional Requirements:

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

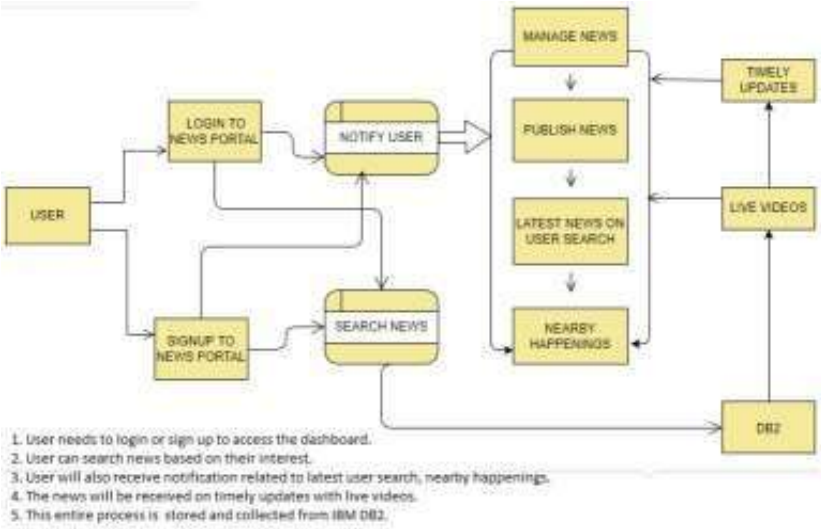
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	➢ The news is updated almost every minute, in all this endless news flow, the user should always be able view exactly information that interest him/her at the moment. ➢ Search and Filtering function will provide the users with the opportunity to search for information of their interest according to specific criteria and parameters.
NFR-2	**Security**	➢ Authentication and password management ➢ During the comment, many users can be rude or Cruel, so of course, it worth thinking of algorithms that will allow you to block offensive comments, as well as spam.
NFR-3	**Reliability**	➢ Avoid the Fake news ➢ Instant news at instant time ➢ Track the location of the news
NFR-4	**Performance**	➢ Keep users In-App for longer with related post ➢ Show users the most relevant story first ➢ Keep your user's attention with list views
NFR-5	**Availability**	➢ Continuous running for example, 24/7, minimum idle time ➢ History of the previous news that happened before related to the present news
NFR-6	**Scalability**	➢ Get more user's by encouraging social sharing ➢ Keep users In-App for longer with related post

`5. PROJECT DESIGN

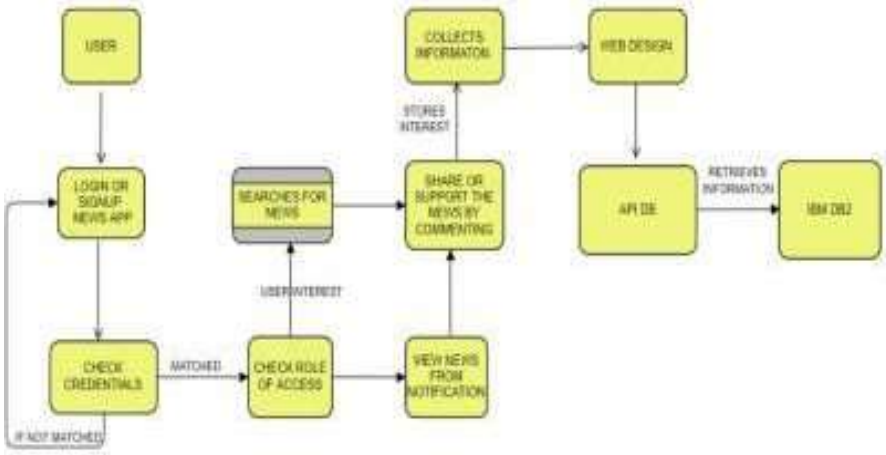
5.1. Data Flow Diagrams:

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 ofthe system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

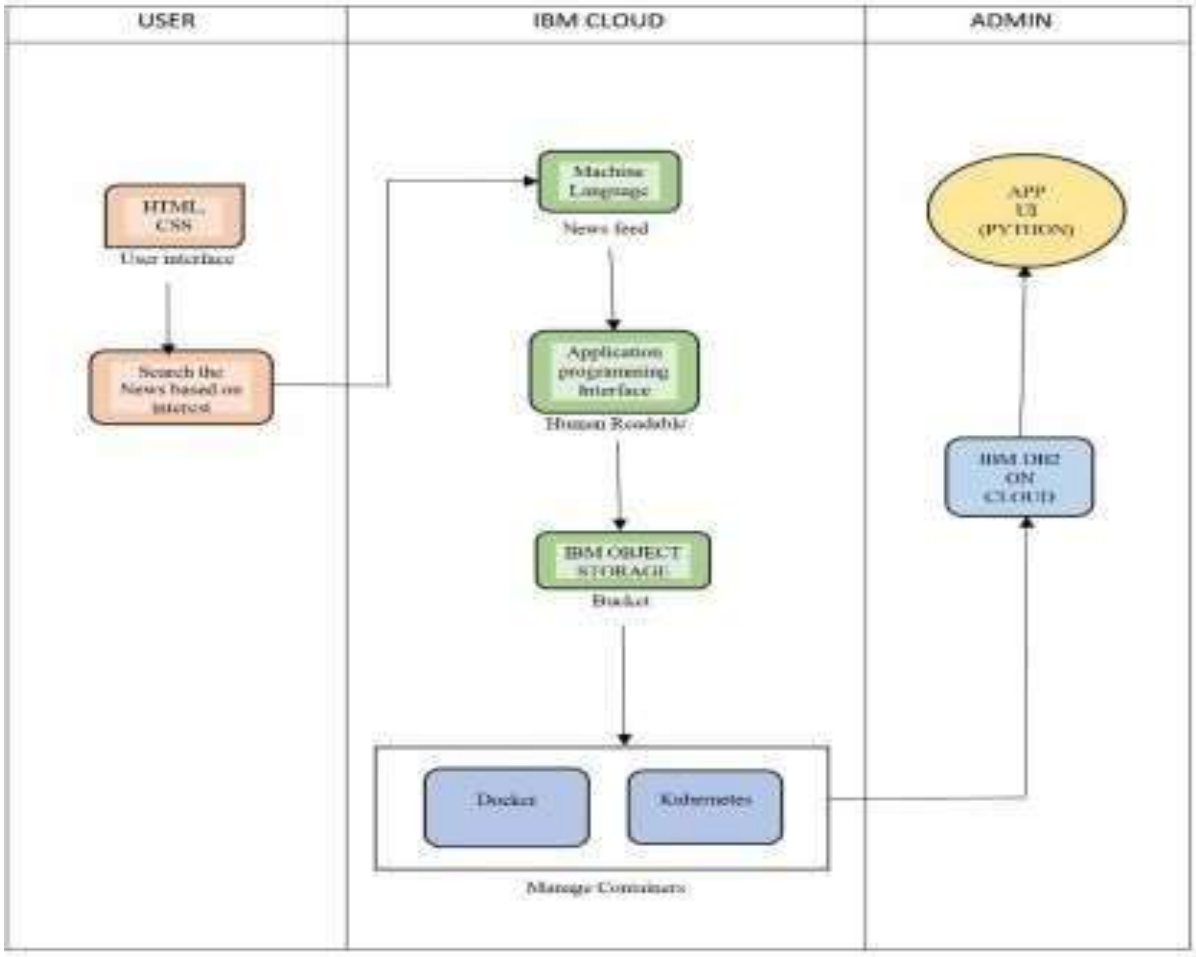
Example: (Simplified)



Example: DFD Level 0(Industry Standard)



5.2. Solution & Techinical Architecture



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 Ihave registered for the application	I can receive confirmationemail & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application throughFacebook	I can register & access thedashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application throughGmail	I can receive password to mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by enteringemail & password	I can receive confirmationmail.	High	Sprint-1
	Dashboard	USN-6	The news portal and shows the recent NEWS as Breaking NEWS	I can Open and view the newsportal	Low	Sprint-2
Client(Web user)	Search Bar	USN-7	User searches for news based on their own interest	I can view the related newsand can watch videos	High	Sprint-1
		USN-8	The news can be viewed that is appearing on the dashboard.	I shall click on the news wanted and can open it.	High	Sprint-1
Administrator	Server	USN-9	Provides correct news available from the database.		Medium	Sprint-1
		USN-10	Provide live news with video and audio contents	I can get the news in which i'minterested.	High	Sprint-1

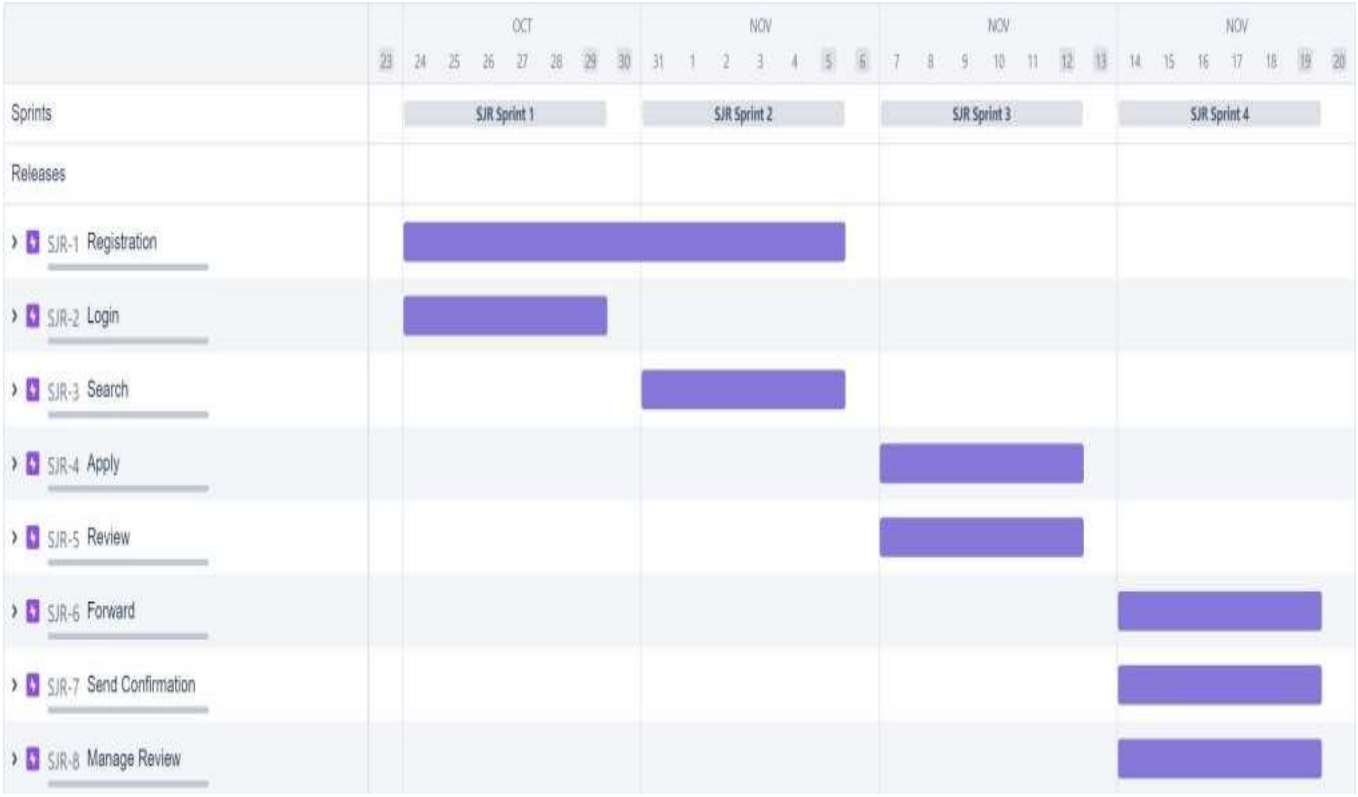
6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	2	High	Ganesh Kumar
Sprint-1	Register	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Vishwa Vengadesh
Sprint-1	Profile	USN-3	As a user, I can register for the application through Facebook	2	Medium	Sathish
Sprint-1	Profile	USN-4	As a user, I can register for the application through Gmail	2	Medium	Tamil selvan
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Gokula krishnan
Sprint-1	Dashboard	USN-6	As a user I should be able to navigate and access all the features hassle free	2	High	Sathish, Tamil selvan
Sprint-1	Layout	USN-7	As a user I should be able to access the portal with different devices with the same comfort	2	High	Gokula krishnan
Sprint-1	Data Store and retrieval	USN-8	Get Data from API and store as JSON in DB2	3	High	Vishwa Vengadesh, Ganesh Kumar
Sprint-1	Data Store and retrieval	USN-9	Get bin data from API and store in DFS	2	High	Vishwa Vengadesh
Sprint-1	User Segregation and data access	USN-10	As a CC executive I should be able to uniquely identify the customer and offer help	1	High	Tamil selvan
Sprint-1	Change code	USN-11	As a administrator I should be able to modify code	2	High	Sathish

			according to the future requirements.			
Sprint-1	Monitor the system	USN-12	As a administrator I should be able to monitor the cloud system and fix errors before customer.	1	High	Ganesh Kumar

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	6 Days	29 Oct 2022	30 Oct 2022	8	02 Nov 2022
Sprint-1	20	6 Days	01 Nov 2022	09 Nov 2022	4	05 Nov 2022
Sprint-1	20	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-1	20	6 Days	14 Nov 2022	19 Nov 2022	4	19 Nov 2022

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

Voice Recognition:

We can search through the voice input also. So, that user can use their mobile device single handedly. No need to type the words. Mic button was placed after the search bar.

Coding: let mic = document.getElementById("mic"); let searchinput = document.getElementById("searchinput");

```
// vibrate
function
vibrate(ms
) {
  navigator.vibrate(ms);
}


function runSpeechRecognition() {
  vibrate(100);

  let recognition = new webkitSpeechRecognition();
  // let recognition = new SpeechRecognition();
  recognition.onstart = () => {
     // toast
  Toastify({
    text: "We are listening you !",
    duration: 2000,
    newWindow: true,
    gravity: "bottom", // `top` or `bottom`
   position: "center", // `left`, `center` or `right`
```

```
    stopOnFocus: true, // Prevents dismissing of toast on hover
style: {
      background: "linear-gradient(to right, #00b09b, #96c93d)",
    },
    onClick: function(){} // Callback after click
  }).showToast();

}
;
  recognition.onresult = (event) => {
    var transcripts = event.results[0][0].transcript;
    console.log(transcripts);
    searchinput.value = "";
    searchinput.value = transcripts;

}
;
  recognition.onspeechend = () => {
    recognition.stop();
            // toast
  Toastify({
    text: "Speech recognition ended",
    duration: 4000,
    newWindow: true,
    gravity: "bottom", // `top` or `bottom`
    position: "center", // `left`, `center` or `right`
    stopOnFocus: true, // Prevents dismissing of toast on hover
     style: {
        background: "linear-gradient(to right, #00b09b, #96c93d)",
     },
```

```
   onClick: function(){} // Callback after click
 }).showToast();

}
;
  recognition.start();

}



searchinput.addEventListener('keypress', function (e) {
  if (e.key === 'Enter') {
    // code for enter
    if(!navigator.onLine){
        Toastify({
          text: "You are offline",
          duration: 4000,
          newWindow: true,
          gravity: "bottom", // `top` or `bottom`
          position: "center", // `left`, `center` or `right`
          stopOnFocus: true, // Prevents dismissing of toast on hover
          style: {
            background: "linear-gradient(to right, #00b09b, #96c93d)",
          },
          onClick: function(){} // Callback after click
        }).showToast();
      }
  }
});
```

7.2 Feature 2:

Chat-Bot:

Watson Assistant Chatbot

```
window.watsonAssistantChatOptions = {
  integrationID: "a6d7e889-59ed-46da-8168-775bffd4611e", // The ID of this integration.
  region: "us-east", // The region your integration is hosted in.
  serviceInstanceID: "97212d7f-a694-4baf-a9a3-40807857702a", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://webchat.global.assistant.watson.appdomain.cloud/versions/"+(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
});
```

8. TESTING :

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.Test Management is a collaborative, quality management solution that offers end-to-end test planning and test asset management, from requirements to defects. Teams can seamlessly share information and use automation to speed complex project schedules and report on metrics in real time for informed release decisions.

8.1 Test Cases:

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

8.2 User Acceptance Testing :

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).

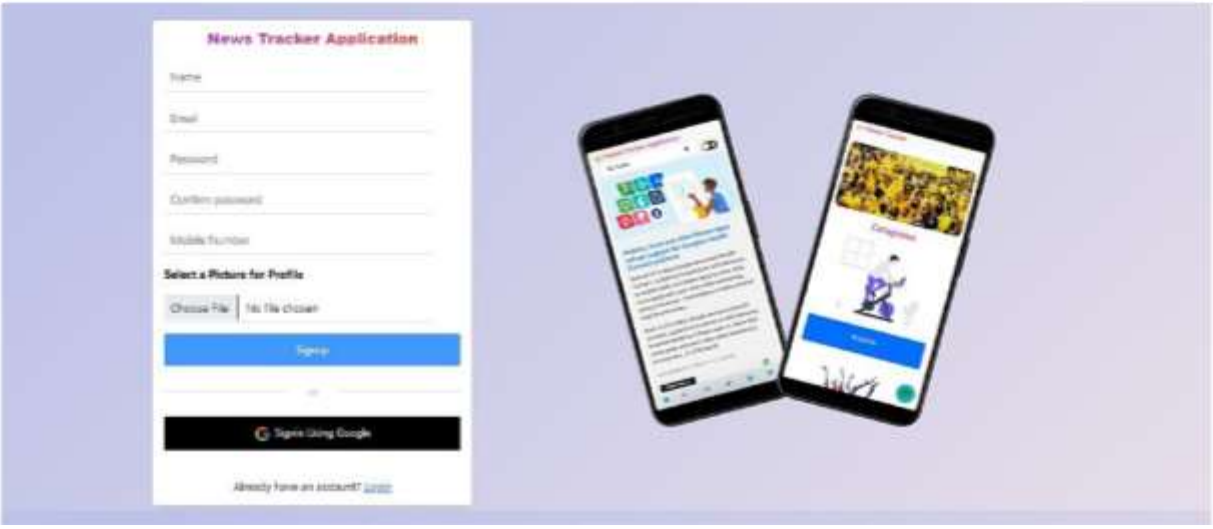
Defect Analysis: This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

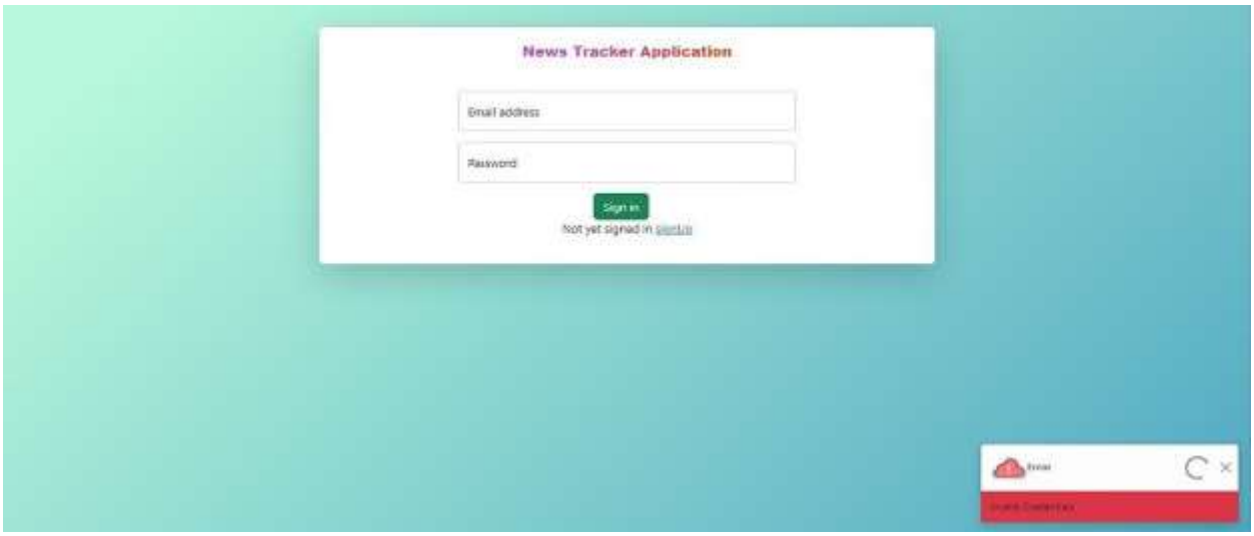
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	1	3	1	6
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	80

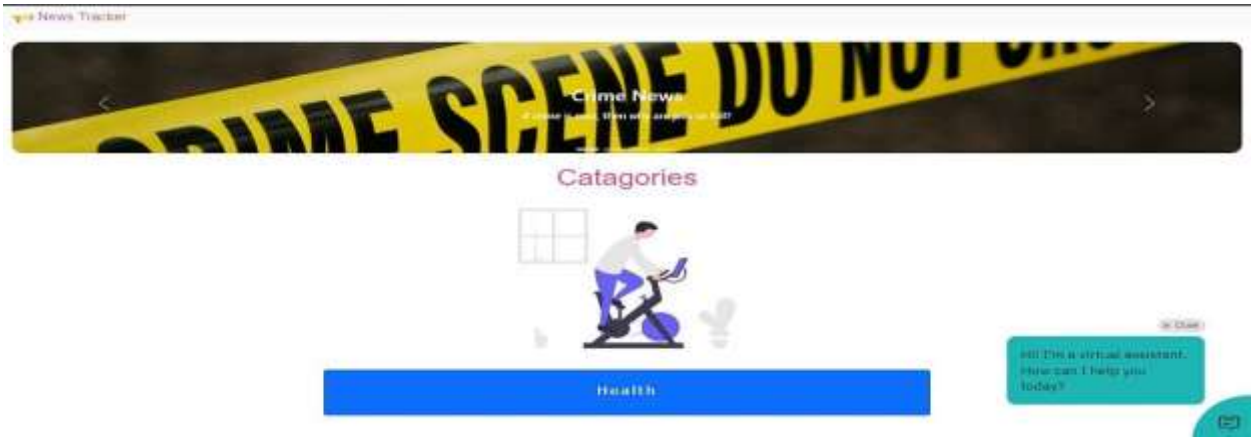
Test Case Analysis: This report shows the number of test cases that have passed, failed, and untested.

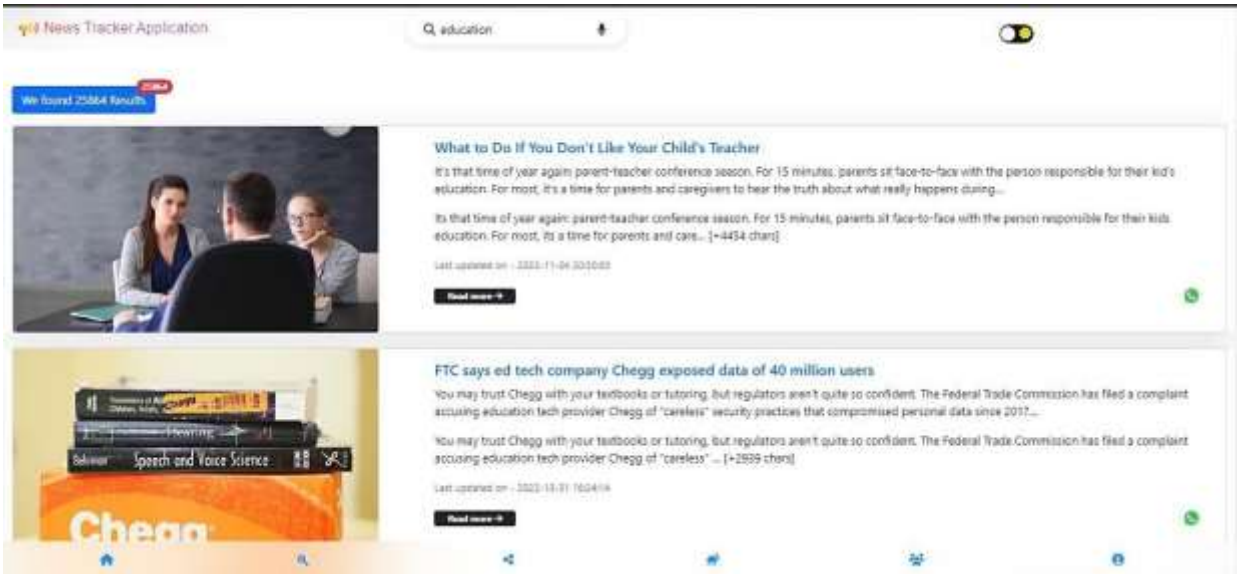
Outsource Shipping	3	0	1	3
Exception Reporting	9	0	1	9
Final Report Output	4	0	1	4
Version Control	2	0	0	2

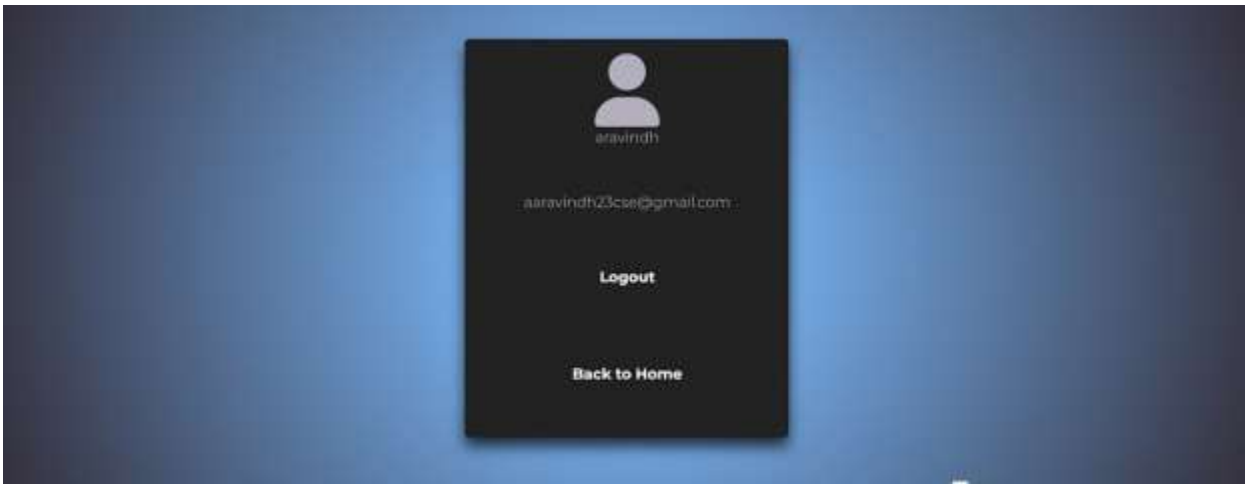
9. RESULTS :




News Tracker Application
Email address
Password
Sign in
Not yet signed in sign Up


Crime News
Catagories
Health


News Tracker Application
education
What to Do If You Don't Like Your Child's Teacher
FTC says ed tech company Chegg exposed data of 40 million users



10. ADVANTAGES & DISADVANTAGES:

ADVANTAGES

- Real-time application is allowed and has live layout.
- It has a Built-in support for Google Platform.
- News accuracy of specialized API Authenticated users are allowed Up to date and daily information is available

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.

11. CONCLUSION:

In our project work, an attempt has been made to develop a News or information-based website. We develop this project that helps people and make them aware so that they can know any news. To establish this website we use various methodologies. To develop this project we faced many problems but we hardly tried to develop this project. The project we've offered here is just the beginning of a new way of interacting with our society. In the meantime, don't forget that compelling visual content will help you be more visible and viral than offline or online newspapers.

12. FUTURE SCOPE:

We are in a process of developing a algorithm that will help the user to read the new postings and news from his recent data sources. In pandemic situations, offline news won't be delivered to anyone, in those time these news apps are the most suitable. In the future, we are going to develop a new categories according to their user locations.

13. APPENDIX:

Source Code:

```
import json
import bcrypt
import ibm_db
import requests
from flask import (Flask, redirect, render_template, request)

app =  Flask(__name__)

# ========================================  for database with
IBM==========================
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pxb18488;PWD=LsK2CkZjU8qTNHm
V",'','')
print(conn)
print("🔐 connection successfull with IBM_DB ⚡")

# signup form data
@app.route('/')
def index():
  return render_template('signup.html')
```

```
# signup form validation
@app.route('/signUpFormData',methods = ['POST', 'GET'])
def signUpFormData():
        if request.method == "POST":
            userName = request.form.get("userName",False)
            userEmail = request.form.get("userEmail")
            userPassword = request.form.get("userPassword")
            userConfirmPassword = request.form.get("userPasswordConfirm")

            if userPassword == userConfirmPassword:
                sql = "SELECT * FROM news_tracker_application WHERE
userEmail =?"
                stmt = ibm_db.prepare(conn, sql)
                ibm_db.bind_param(stmt,1,userEmail)
                ibm_db.execute(stmt)
                account = ibm_db.fetch_assoc(stmt)
                # print(account)

                bytes = userPassword.encode('utf-8')

                salt = bcrypt.gensalt()

                hashed_password = bcrypt.hashpw(bytes, salt)

                userPassword = hashed_password


                if account:
                    return render_template('login.html', msg="You are
already a member, please login using your details")
                else:
                    insert_sql = "INSERT INTO news_tracker_application
VALUES (?,?,?)"
                    prep_stmt = ibm_db.prepare(conn, insert_sql)
                    ibm_db.bind_param(prep_stmt, 1, userName)
                    ibm_db.bind_param(prep_stmt, 2, userEmail)
                    ibm_db.bind_param(prep_stmt, 3, userPassword)
                    ibm_db.execute(prep_stmt)


                    # from sendgrid import SendGridAPIClient
                    # from sendgrid.helpers.mail import Mail

                    # message = Mail(
```

```
                            #        from_email='applicationnewstracker@gmail.com
',
                            #        to_emails=userEmail,
                            #        subject='Welcome to News Tracker
Application',
                            #        html_content='<img src="https://cloud-
object-storage-18-cos-standard-yx0.s3.jp-tok.cloud-object-
storage.appdomain.cloud/welcom_nta.gif" />')
                            # try:
                            #        sg =
SendGridAPIClient('SG.29Td0tbNSkyliF9SSPnQNA.4DBECk8ka8RmmYRE5OIsRKGOR2QI2raRG3CL
mdsVBVc')
                            #        response = sg.send(message)
                            #        print(response.status_code)
                            #        print(response.body)
                            #        print(response.headers)
                            # except Exception as e:
                            #        print(str(e))

                            return render_template('login.html', msg="user Data
saved successfuly.. Please login use your credentials")

              else:
                     return render_template('signup.html', msg = 'Password and
Confirm Password are not matched' )

# ============================================= for
serve  ====================================

# login form validation
@app.route('/loginForm', methods=['GET', 'POST'])
def loginForm():
    if request.method == 'POST':

        global email
        email = request.form['userEmail']
        pwd = request.form['userPassword']

        var = email

        sql = "SELECT * FROM news_tracker_application WHERE userEmail =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        auth_token = ibm_db.fetch_assoc(stmt)
```

```
        print("auth",auth_token)

        if auth_token:
            # encoding user password
            userBytes = pwd.encode('utf-8')
            byte_pwd = bytes(auth_token['USERPASSWORD'], 'utf-8')

            # checking password
            result = bcrypt.checkpw(userBytes, byte_pwd)

            if result:
                print("succ")
                url = (' https://newsapi.org/v2/top-
headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
                TopHeadlinesResponse = requests.get(url).json()
                return render_template('index.html', msg="Logged in
Successfully", responseData=TopHeadlinesResponse, tmp = 1)
            else:
                return render_template('login.html', msg="Invalid Credentials",
tmp = 0)
        else:
            return render_template('signup.html', msg="User doesn't exist, Please
Register using your details!")
    else:
        return render_template('login.html', title='Sign In')

# home page
@app.route('/home')
def userdata():
       print(email)
       url = (' https://newsapi.org/v2/top-
headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
       TopHeadlinesResponse = requests.get(url).json()
       return render_template('index.html',responseData=TopHeadlinesResponse)

# signup form
@app.route('/')
@app.route('/signup')
def signUp():
       return render_template('signup.html')

# login form
@app.route('/login')
def login():
       return render_template('login.html')
```

```
# logout
@app.route('/logout')
def logout():
       return redirect('/login')

# redirect Home
@app.route('/redirectHome')
def redirectHome():
       return redirect('/home')

# about us
@app.route('/aboutus')
def aboutus():
       return render_template('aboutus.html')

# education
@app.route('/education')
def education():
       value = 'education'
       crimenews = ('https://newsapi.org/v2/everything?'
'q='+value+'&''from=2022-10-
29&''sortBy=popularity&''apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
       educationResponse = requests.get(crimenews).json()
       print(educationResponse)
       # return
render_template('NewsTemplate.html',responseData=crimeNewsresponse)   dharun API
key = 7c7062c3a98649b5bc6ffda7fdc5a01b aravindh =
9b6f57afe98440b8b362b1046559d71d
       result_count = educationResponse.get('totalResults')
       if(result_count>0):
          return
render_template('NewsTemplate.html',responseData=educationResponse,returned_input
_search_value=value,result_count=result_count)
       else:
          return render_template('notfound.html')

# Top headlines
@app.route('/TopHeadlines')
def TopHeadlines():
       value ='Top Headlines'
       url = (' https://newsapi.org/v2/top-
headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
       TopHeadlinesResponse = requests.get(url).json()
       result_count = TopHeadlinesResponse.get('totalResults')
```

```
        return
render_template('NewsTemplate.html',responseData=TopHeadlinesResponse,returned_in
put_search_value=value,result_count=result_count)

# science news
@app.route('/sciencenews')
def crimenews():
        value ='science'
        sciencenews = ('https://newsapi.org/v2/everything?'
        'q='+value+'&'
        'from=2022-10-29&'
        'sortBy=popularity&'
        'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
        scienceNewsresponse = requests.get(sciencenews).json()
        print(scienceNewsresponse)
        #    dharun API key = 7c7062c3a98649b5bc6ffda7fdc5a01b aravindh =
9b6f57afe98440b8b362b1046559d71d
        result_count =scienceNewsresponse.get('articles')
        result_count = len(result_count)

        if(result_count>0):
           return
render_template('NewsTemplate.html',responseData=scienceNewsresponse,returned_inp
ut_search_value=value,result_count=result_count)
        else:
           return render_template('notfound.html')

# health news
@app.route('/healthnews')
def healthnews():
        value = 'health'
        healthnews = ('https://newsapi.org/v2/everything?'
         'q='+value+'&'
        'from=2022-10-29&'
        'sortBy=popularity&'
        'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
        healthNewsresponse = requests.get(healthnews).json()
        result_count = healthNewsresponse.get('totalResults')
        if(result_count>0):
           return
render_template('NewsTemplate.html',responseData=healthNewsresponse,returned_inpu
t_search_value=value,result_count=result_count)
        else:
           return render_template('notfound.html')
```

```
# sports news
@app.route('/sportsnews')
def sportsnews():
       value = 'sports'
       sportsnews = ('https://newsapi.org/v2/everything?'
       'q='+value+'&'
       'from=2022-10-29&'
       'sortBy=popularity&'
       'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')
       sportsNewsresponse = requests.get(sportsnews).json()
       # return
render_template('NewsTemplate.html',responseData=crimeNewsresponse)
       result_count = sportsNewsresponse.get('totalResults')
       if(result_count>0):
          return
render_template('NewsTemplate.html',responseData=sportsNewsresponse,returned_inpu
t_search_value=value,result_count=result_count)
       else:
          return render_template('notfound.html')

@app.route('/searchResults', methods =["POST"])
def searchResults():
       if request.method == "POST":
              search_value_name = request.form.get("searchvalue")

              print(search_value_name)

              searchURL = ('https://newsapi.org/v2/everything?'
              'q='+search_value_name+'&'
              'from=2022-10-29&'
              'sortBy=popularity&'
              'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

              searchResponse = requests.get(searchURL).json()
              result_count = searchResponse.get('totalResults')

              print(result_count)  # NUMBER

              if(result_count>0):
                 return
render_template('NewsTemplate.html',responseData=searchResponse,returned_input_se
arch_value=search_value_name,result_count=result_count)
              else:
```

```
                return
render_template('notfound.html',responseData=searchResponse,returned_input_search
_value=search_value_name)

# tab user
@app.route('/tabuser')
def tabuser():

       userEmail = email
       print('email',userEmail)
       sql = "SELECT * FROM news_tracker_application WHERE userEmail =?"
       stmt = ibm_db.prepare(conn, sql)
       ibm_db.bind_param(stmt, 1, userEmail)
       ibm_db.execute(stmt)
       auth_token = ibm_db.fetch_assoc(stmt)

       return render_template('userinfo.html', msg=auth_token)

# logout
@app.route('/logout')
def logoutform():
       email = ''
       return render_template('login.html', msg= 'successfully logged out')

#================================= server details
====================================

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

GitHub:
https://github.com/IBM-EPBL/IBM-Project-44272-1660723624