

# **NEWS TRACKER APPLICATION**

BACHELOR OF TECHNOLOGY  
IN  
**INFORMATION TECHNOLOGY**

Submitted By

**Team ID: PNT2022TMID25308**

VIJAYALAKSHMI S	210619205052
SHARMILA P	210619205043
JENITA GRACELIN J	210619205020
MAHALAKSHMI J	210619205026



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JEPPIAAR INSTITUTE OF TECHNOLOGY  
CHENNAI**

## **ABSTRACT :**

We have quick connections and networks that allow us to connect to other people right away as global technology is developing so quickly. Most people currently have access to daily use devices like mobile phones, tablets, and laptops. We must keep up with all happenings and news in our quick and information-driven environment. Users of this Android news app can access the most recent news from 120+ newspapers in 50+ countries. This application's major goal is to connect news stories from all around the world and give them to users as quickly as possible in the finest visual format.

# TABLE OF CONTENT

## 1. INTRODUCTION

- a. Project Overview
- b. Purpose

## 2. LITERATURE SURVEY

- a. Existing problem
- b. References
- c. Problem Statement Definition

## 3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

## 4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

## 5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

## 6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule

## 7. CODING & SOLUTION

- a. Feature 1
- b. Feature 2

## 8. TESTING

- a. Test Cases

## 9. RESULTS

## 10. CONCLUSION

## 11. FUTURE SCOPE

## 12. APPENDIX

Source Code GitHub & Project Demo Link

# **NEWS TRACKER APPLICATION**

## **1. INTRODUCTION**

This app is a news app. This Application will display global news. Since you can now view news wherever you are, this is superior than old-fashioned newspapers. Live updates will be provided for this. As an editor, you can add your own news to this app, and other users can view your published news as well.

## **a. PROJECT OVERVIEW**

The project's goal is to create an application that will show news items and videos that have been reviewed and approved at the back end. The user can search through many categories of news articles and videos. User will be able to view news details, photographs, and video using the app. Any news story or video can be bookmarked by users for later use. At the backend of the programme, there will be three different types of users. The admin manager will have complete access to all backend capabilities, including user and content management. The only other users who will have limited access to backend functions are reporters and editors. While reporters will be responsible for writing news articles and producing news videos, editors will be in charge of approving and publishing the reporters' articles and films of each user.

## **b. PURPOSE**

Reading newspaper might sometimes be time taking considering the size of the article. Sometimes people give the article a skip due to lethargy if it is too long. Information through listening and visuals is quickly perceived compared to reading. To make things easier, the NewsApp android application can be used to scan the headlines from the newspaper and fetch relevant videos from the world wide web.

## **2. LITERATURE SURVEY**

### **a. EXISTING PROBLEM**

We frequently believe that there aren't enough hours in the day to get everything done. That's not realistic, but you can cut down on the time by reading the news differently than you usually do.

### **b. REFERENCES**

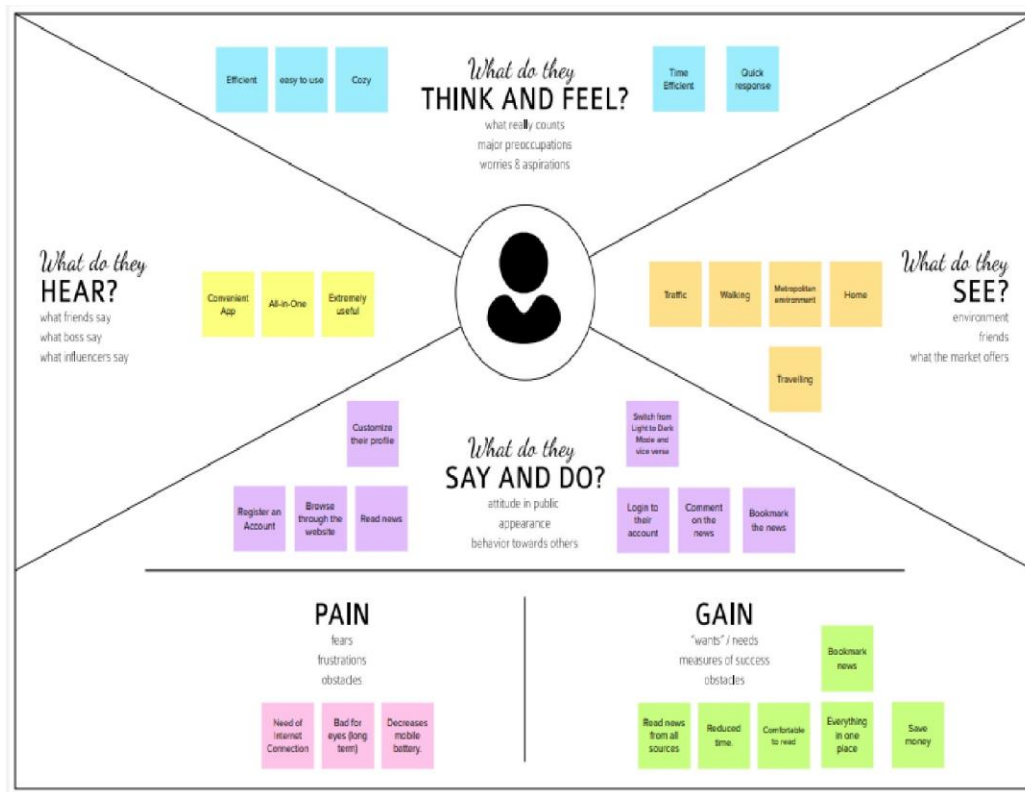
[https://www.researchgate.net/publication/299870645\\_Exploring\\_mobile\\_news\\_reading\\_interactions\\_for\\_news\\_app\\_personalisation](https://www.researchgate.net/publication/299870645_Exploring_mobile_news_reading_interactions_for_news_app_personalisation)

### **c. PROBLEM STATEMENT DEFINITION**

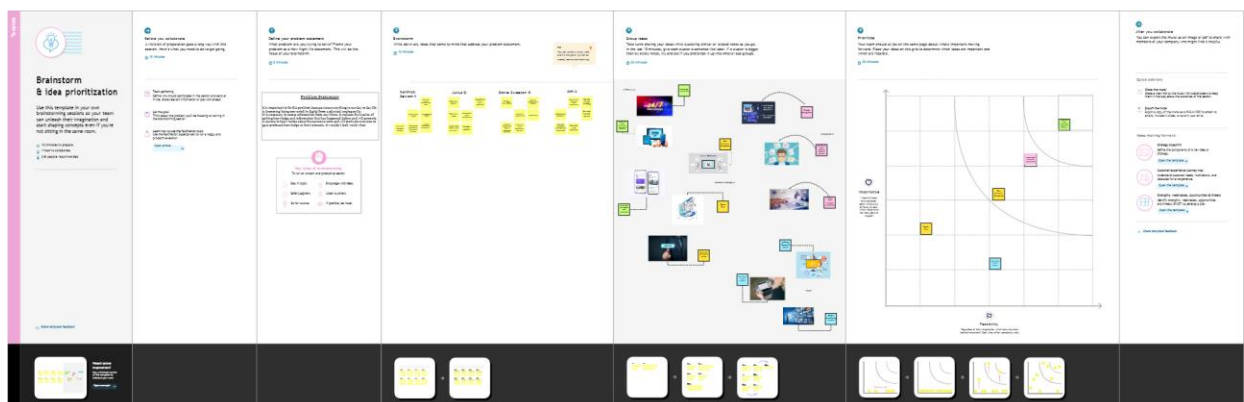
- > The manual newspaper system entails all of the clerical work, which is a laborious and difficult task. Users find it challenging to access news on the go in today's fast-paced world.
- > The user wants a solution to manage his daily news intake so that he can keep up with happenings while still managing his busy schedule.
- > In order to stay informed about his interests, the user wishes to read news just about specific themes.
- > The user wants to read just reputable news sources for their information.
- > The user requires a mechanism to do news searches on the subjects they choose.
- > The user uses GPS position to learn news about his neighbourhood.

# 3. IDEATION & PROPOSED SOLUTION

## a. EMPATHY MAP CANVAS



## b. IDEATION & BRAINSTORMING



### **c. PROPOSED SOLUTION**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The majority of people dislike carrying around a newspaper. Some people want to just receive updates on the topics they are interested in.
2.	Idea / Solution description	Users should be able to tailor their region of interest and receive news whenever they want thanks to an application that is currently being developed. in order for them to be informed when any relevant new information is updated.
3.	Novelty / Uniqueness	<ol style="list-style-type: none"><li>1. Rather than reading all the news, a user can read the news only in the fields in which they are interested.</li><li>2. This program offers consumers access to a safe and reliable ecosystem. The application only shares authentic, spam-free news .</li></ol>
4.	Social Impact / Customer Satisfaction	Users of this program are encouraged to provide comments. Eventually, changes were made in response to that feedback..
5.	Business Model (Revenue Model)	<ol style="list-style-type: none"><li>1. We should include advertisements in the application so that we can receive funding from companies that sponsor the ads. Additional advertising could annoy the user..</li><li>2. Add premium subscription, users who subscribe for premium won't get advertisements.</li></ol>
6.	Scalability of the Solution	As it was an application-based project, good ideation and execution can result in the development of a flawless application. If the user likes our application, they may recommend and share it with others, which would boost our application insights.



## d. PROBLEM SOLUTION FIT

Project Title: News Tracker Application		Project Design Phase-I - Solution Fit Template		Team ID: PNT2022TMID53854	
Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> 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.	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> 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	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> 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	
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> 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	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> 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	<b>7. BEHAVIOUR</b> <b>BE</b> 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, fit into CC	<b>3. TRIGGERS</b> <b>TR</b> People asking about latest news When things goes viral When need report about weather, market, sports and etc	<b>10. YOUR SOLUTION</b> <b>SL</b> 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.	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> 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	
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> Feels waste of time to read irrelevant content Feels frustrated about misleading news Feels getting lack of information from contents				



## **4. REQUIREMENT ANALYSIS**

### **a. FUNCTIONAL REQUIREMENT**

Following are the functional requirements of the proposed solution.

<b>FR No.</b>	<b>Functional Requirement (Epic)</b>	<b>Sub Requirement (Story / Sub-Task)</b>
FR-1	User Registration	Registration through online application Registration through Gmail Registration through website
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login through browser directly by entering username and password Login through Login through email
FR-4	User interaction	Done through user interface between client and server View the related news by subscribed or requested page

### **b. NON-FUNCTIONAL REQUIREMENTS**

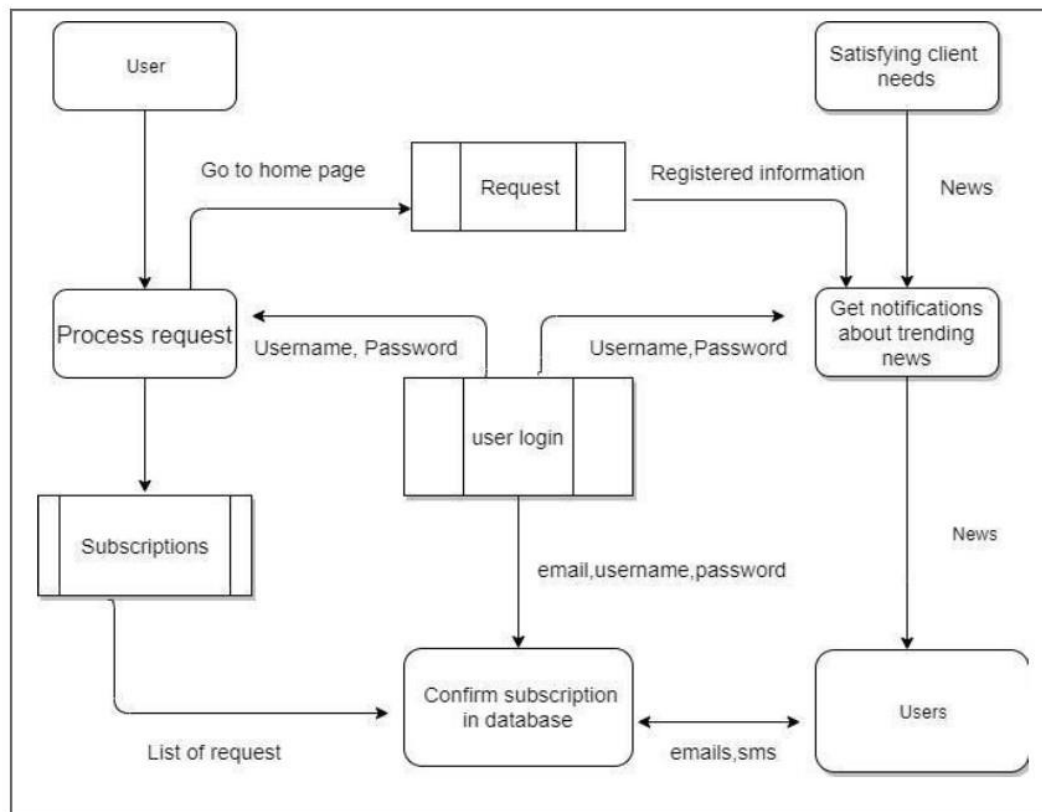
Following are the non-functional requirements of the proposed solution.

<b>FR No.</b>	<b>Non-Functional Requirement</b>	<b>Description</b>
NFR-1	<b>Usability</b>	By subscribing to the website's news feed, end users can receive push notifications for new information on the site.
NFR-2	<b>Security</b>	How well are the system and its data protected against attacks
NFR-3	<b>Reliability</b>	How frequently do the system's critical failures occur? How long does it take to resolve the problem once it occurs? And how does downtime compare to user availability time?
NFR-4	<b>Performance</b>	<p>The primary non-functional criteria that every system must have is performance. It specifies how quickly a software system or a specific component of it reacts to specific user actions while handling a specific workload. Given the current user base as a whole, this statistic often indicates how long a user must wait before the goal operation occurs (the page renders, a transaction is executed, etc.).</p> <p>But it isn't always the case.</p>

		Performance specifications could list unnoticed by users background tasks like backup. Let's instead concentrate on user-centric performance..
NFR-5	<b>Availability</b>	Availability refers to the likelihood that a user will be able to use the system at a specific time. You can define it as a percentage of the time the system is available for operation within a given time period, while it can also be represented as an expected percentage of requests that are successful. For instance, during a month, the system might be accessible 98% of the time. Perhaps the most important business requirement is availability, but to define it, you also need to have estimates for dependability and maintainability..
NFR-6	<b>Scalability</b>	Scalability measures the highest workloads that the system can handle while still delivering the required levels of performance. When workloads increase, your system can grow vertically or horizontally using one of two methods.

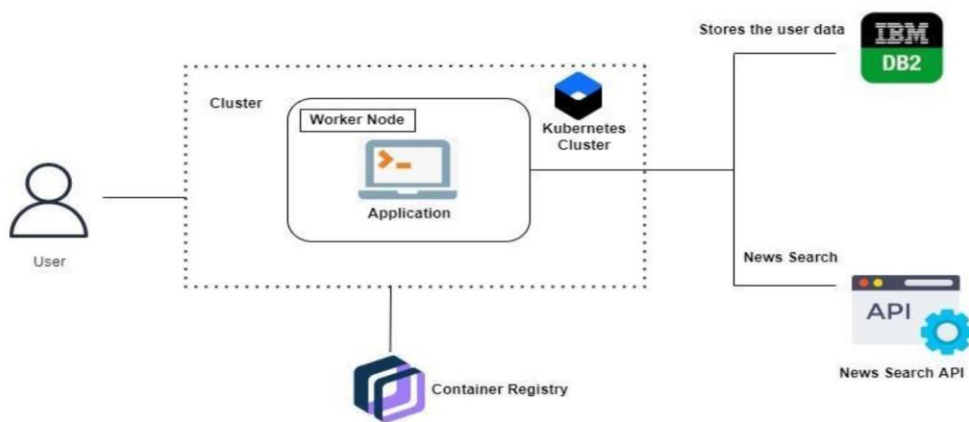
## 5.PROJECT DESIGN

### a. DATA FLOW DIAGRAM



## **b. SOLUTION & TECHNICAL ARCHITECTURE**

Technical Architecture:



## **c. USER STORIES**

### **User Stories**

Use the below template to list all the user stories for the product.

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Searching news)	Registration	USN-1	As a user, I can register to the application by entering my email, 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.	High	Sprint-1
		USN-3	As a user, I can register for the application through their given website	I can register & access the dashboard with login.	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can view all types of information through this application	High	Sprint-1
	Dashboard	USN-6	To see their histories about recently viewed, updates for search related news, current progress, feedback			Sprint-1

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Browser	USN-7	Have interactive medium between client and server	I have a clarity to use this application and easily resolve my specific issues	high	Sprint-1
Customer Care Executive	Chat Bot	USN-8	Rectify the issues related Subscription, account, terms and conditions, privacy policy			Sprint-2

## **6.PROJECT PLANNING & SCHEDULING**

### ***a.SPRINT PLANNING & ESTIMATION***

TITLE	DESCRIPTION	DATE
Literature Survey	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	19 Oct 2022
Empathy map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	19 Oct 2022
Ideation	List the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	19 Oct 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	27 Oct 2022
Problem Solution Fit	Prepare problem solution fit document.	27 Oct 2022
Solution Architecture	Prepare solution architecture document.	27 Oct 2022
Customer Journey	Prepare the customer journey maps to understand the user	27 Oct 2022



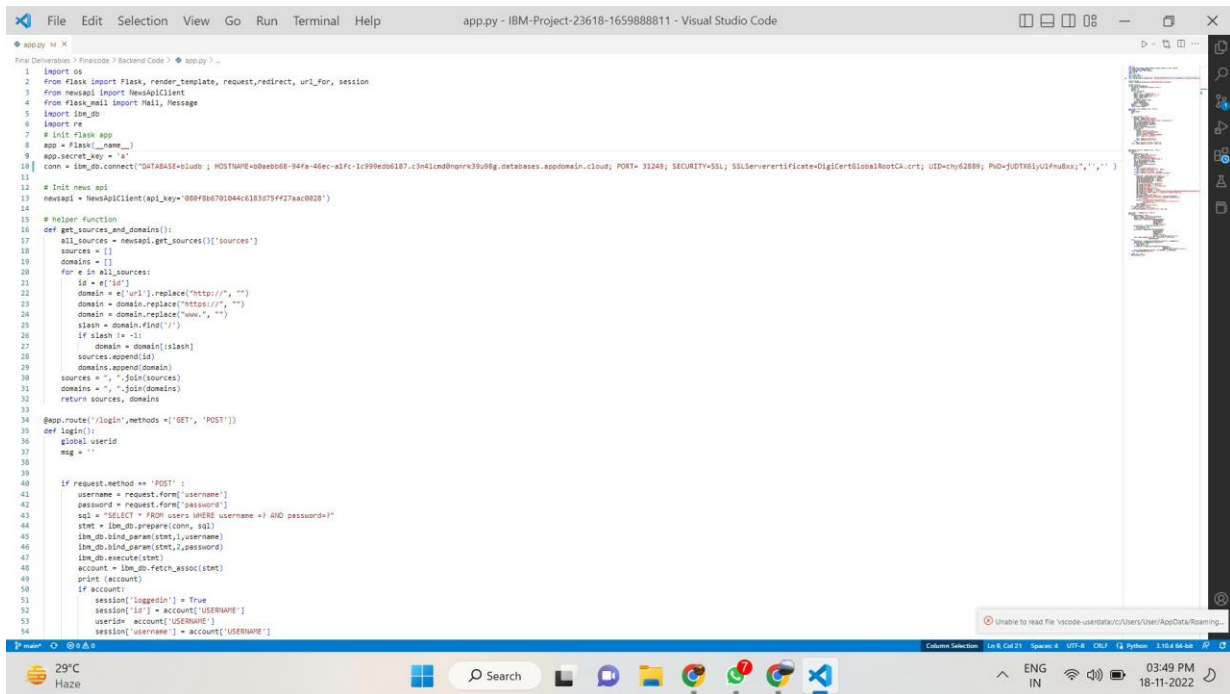
	interactions & experiences with the application (entry to exit).	
Functional Requirement	Prepare the functional requirement document.	27 Oct 2022
Data Flow Diagrams	Prepare the data flow diagrams	27 Oct 2022
Technology Architecture	Prepare the technology architecture diagram.	27 Oct 2022
Milestone and Activity List	Prepare the milestones & activity list of the project.	2 Nov 2022
Sprint Planning	Prepare the sprint delivery plan	2 Nov 2022
Project Development	Develop & test the features of the application.	In progress

## **b. SPRINT DELIVERY SCHEDULE**

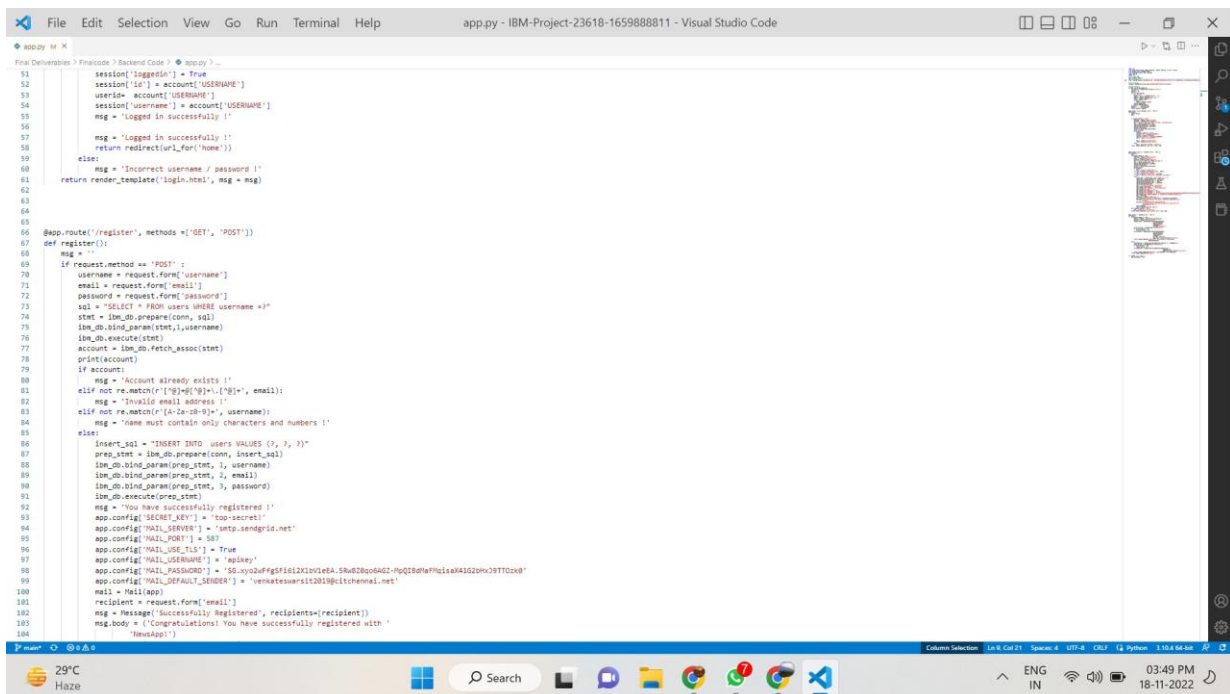
Use the below template to create product backlog and sprint schedule

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 my password.	2	High	Julius ,Karthick
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Donie Sweeton , Athi
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Medium	
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Karthick ,Julius
	Dashboard		As a user, I can search and get suggestions on current news feeds	2	High	Donie Sweeton.
Sprint-3	Notification	USN-6	As a user, I can receive notifications on trending and subscribed news articles	2	Medium	
Sprint-4	Updates	USN-7	As a user, I can get updated and latest news whenever I use this service	2	High	Karthick ,Julius, Donie

# 7.CODING & SOLUTIONS



```
1 import os
2 from flask import Flask, render_template, request, redirect, url_for, session
3 from newapi import NewApiClient
4 from flask_mail import Mail, Message
5 import im_db
6 import re
7 # Init Flask app
8 app = Flask(__name__)
9 app.secret_key = 'a'
10 conn = im_db.connect(("DATABASE=cloud; HOSTNAME=db000008-94fe-4dec-af3c-1c99ed0187.c104indbqprk39d8g.databases.appdomain.cloud; PORT= 31249; SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=cnq42889; PID=jdT86y0fufu0xq;",""))
11
12 # Init new api
13 newapi = NewApiClient(api_key="000F0670104C6183d75f27aac0828")
14
15 # helper function
16 def get_sources_and_domains():
17     all_sources = newapi.get_sources()['sources']
18     sources = []
19     domains = []
20     for s in all_sources:
21         id = s['id']
22         domain = s['url'].replace("http://", "")
23         domain = domain.replace("https://", "")
24         domain = domain.replace("www.", "")
25         slash = domain.find("/")
26         if slash != -1:
27             domain = domain[:slash]
28         sources.append(id)
29         domains.append(domain)
30     sources = ", ".join(sources)
31     domains = ", ".join(domains)
32     return sources, domains
33
34 @app.route("/login", methods=['GET', 'POST'])
35 def login():
36     global userId
37     msg = ""
38
39     if request.method == 'POST':
40         username = request.form['username']
41         password = request.form['password']
42         sql = "SELECT * FROM users WHERE username =? AND password=?"
43         stmt = im_db.prepare(conn, sql)
44         im_db.bind_param(stmt, 1, username)
45         im_db.bind_param(stmt, 2, password)
46         im_db.execute(stmt)
47         account = im_db.fetch_assoc(stmt)
48         print(account)
49         if account:
50             session['loggedin'] = True
51             session['id'] = account['USERID']
52             session['username'] = account['USERNAME']
53             msg = "Logged in successfully !"
54             return redirect(url_for('home'))
55         else:
56             msg = "Incorrect username / password !"
57             return render_template("login.html", msg = msg)
58
59 @app.route("/register", methods=['GET', 'POST'])
60 def register():
61     msg = ""
62
63     if request.method == 'POST':
64         username = request.form['username']
65         email = request.form['email']
66         password = request.form['password']
67         sql = "SELECT * FROM users WHERE username =?"
68         stmt = im_db.prepare(conn, sql)
69         im_db.bind_param(stmt, 1, username)
70         im_db.execute(stmt)
71         account = im_db.fetch_assoc(stmt)
72         print(account)
73         if account:
74             msg = "Account already exists !"
75         elif not re.match(r"^[a-z0-9]+@[a-z0-9]+\.[a-z0-9]+", email):
76             msg = "Invalid email address !"
77         elif not re.match(r"^[a-z0-9]+", username):
78             msg = "name must contain only characters and numbers !"
79         else:
80             insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
81             prep_stmt = im_db.prepare(conn, insert_sql)
82             im_db.bind_param(prepare_stmt, 1, username)
83             im_db.bind_param(prepare_stmt, 2, email)
84             im_db.bind_param(prepare_stmt, 3, password)
85             im_db.execute(prepare_stmt)
86             msg = "You have successfully registered !"
87             app.config['SECRET_KEY'] = 'top-secret!'
88             app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
89             app.config['MAIL_PORT'] = 587
90             app.config['MAIL_USE_TLS'] = True
91             app.config['MAIL_USERNAME'] = 'apikey'
92             app.config['MAIL_PASSWORD'] = "SG.0x2dFg5F1612X2v1eEA.5hU2B0u04GZ-PqQ5B0nFqN1s6K4302Dhw39TTC0x8"
93             app.config['MAIL_DEFAULT_SENDER'] = 'venkateswars13019@litchennai.net'
94             mail = Mail(app)
95             recipient = request.form['email']
96             msg = Message("Successfully Registered", recipients=[recipient])
97             msg.body = ("Congratulations! You have successfully registered with "
98                       "Newapi")
```



```
51 session['loggedin'] = True
52 session['id'] = account['USERID']
53 session['username'] = account['USERNAME']
54 msg = "Logged in successfully !"
55
56 msg = "Logged in successfully !"
57 return redirect(url_for('home'))
58
59 msg = "Incorrect username / password !"
60 return render_template("login.html", msg = msg)
61
62 @app.route("/register", methods=['GET', 'POST'])
63 def register():
64     msg = ""
65
66     if request.method == 'POST':
67         username = request.form['username']
68         email = request.form['email']
69         password = request.form['password']
70         sql = "SELECT * FROM users WHERE username =?"
71         stmt = im_db.prepare(conn, sql)
72         im_db.bind_param(stmt, 1, username)
73         im_db.execute(stmt)
74         account = im_db.fetch_assoc(stmt)
75         print(account)
76         if account:
77             msg = "Account already exists !"
78         elif not re.match(r"^[a-z0-9]+@[a-z0-9]+\.[a-z0-9]+", email):
79             msg = "Invalid email address !"
80         elif not re.match(r"^[a-z0-9]+", username):
81             msg = "name must contain only characters and numbers !"
82         else:
83             insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
84             prep_stmt = im_db.prepare(conn, insert_sql)
85             im_db.bind_param(prepare_stmt, 1, username)
86             im_db.bind_param(prepare_stmt, 2, email)
87             im_db.bind_param(prepare_stmt, 3, password)
88             im_db.execute(prepare_stmt)
89             msg = "You have successfully registered !"
90             app.config['SECRET_KEY'] = 'top-secret!'
91             app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
92             app.config['MAIL_PORT'] = 587
93             app.config['MAIL_USE_TLS'] = True
94             app.config['MAIL_USERNAME'] = 'apikey'
95             app.config['MAIL_PASSWORD'] = "SG.0x2dFg5F1612X2v1eEA.5hU2B0u04GZ-PqQ5B0nFqN1s6K4302Dhw39TTC0x8"
96             app.config['MAIL_DEFAULT_SENDER'] = 'venkateswars13019@litchennai.net'
97             mail = Mail(app)
98             recipient = request.form['email']
99             msg = Message("Successfully Registered", recipients=[recipient])
100             msg.body = ("Congratulations! You have successfully registered with "
101                       "Newapi")
```



Fetch news	Functional	dashboard	Verify user can get the news	News will be feed to the app	404 error	Fail	unsuccessful	App integration problem	ATHI & JULIUS
Types of news available in the fetch newspaper	Functional	news page	Types of news available	Weather, Sport, Economy	Hover button will be shown.	Yes	successful	-	DONIE SWEETON & KARTHICK

## **10. RESULT**

News Tracker Application using cloud is developed and executed at the level of completed progress.

## **11. CONCLUSION**

The current system consists entirely on clerical work, which is a laborious and challenging task. Users find it challenging to access news on the go in today's fast-paced world. Because anyone may read the news online whenever they have time, this approach can be quite helpful. We have presented a technique for tracking consumers' news-reading habits. In order to be able to present a complete adaptive mobile news framework that provides automatic personalization of news apps, we will continue to investigate the design of adaptive interfaces in the future.

## **12. 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.

## **13. APPENDIX**

***GITHUB***

**<https://github.com/IBM-EPBL/IBM-Project-55518-1669097984>**