

IBM NALAIYA THIRAN

NEWS TRACKER APPLICATION

TEAM : PNT2022TMID51641

DOMAIN : CLOUD APPLICATION DEVELOPMENT

PROJECT NAME : NEWS TRACKER APPLICATION

BATCH : B8-2A4E

DEPARTMENT : CSE

COLLEGE NAME : MAR EPHRAEM COLLEGE OF
ENGINEERING AND
TECHNOLOGY

TEAM MEMBERS :

TEAM LEADER : AJISHA JEGAL J J

TEAM MEMBER 1 : KEBIYA G

TEAM MEMBER 2 : MERSHEBA FRED L

TEAM MEMBER 3 : ADLIN ANCY D

ANNA UNIVERSITY: CHENNAI 600 025**BONAFIDE CERTIFICATE**

Certified that this report titled “**NEWS TRACKER APPLICATION**” is the bonafide work of **AJISHAJEGAL J J (961419104007), KEBIYA G (961419104032), MERSHEBA FRED L (961419104032), ADLIN ANCY D(961419104005)** who carried out the mini project work study under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other report or dissertation on the basis of which a degree or award was conferred on earlier occasion or any other candidate.

SIGNATURE

Prof. Dr. Dhanya D, M. E, Ph.D.,
HEAD OF THE DEPARTMENT,

Computer Science and
Engineering, Mar Ephraem
College of Engineering and
Technology, Technology,
Elavuvilai - 629171 Elavuvilai - 629171

Internal Examiner

SIGNATURE

Mrs. Janila J, M.E
SUPERVISOR,

Computer Science and
Engineering, Mar Ephraem
College of Engineering and

External Examiner

ACKNOWLEDGEMENTS

First and foremost, we thank **The Almighty God** for endowing us with their immense blessings that helped us in each step of our progress towards the successful completion of this project report.

We thank Chairman His Excellency Most **Rev. Dr. Vincent Mar Paulos**, (Bishop, Catholic Diocese of Marthandam) for his blessings throughout the project. Then we express our gratitude to **Rev. Fr. Josephin Raj, M.S.W**, Correspondent, Mar Ephraem College of Engineering and Technology for working on a great mission and vision and providing excellent infrastructure.

We thank our Principal, **Prof. Dr. A. Lenin Fred, M.E, Ph.D.**, for providing a great guidance which direct us the right way and for his great support, guidance and unfailing supervision all throughout the project. Then, we would like to express our sincere thanks to **Prof. Dr. N. Austin, M.E, Ph.D.**, Director Teaching, Learning and Faculty development, for his valuable encouragements and support during the course.

Especially we would like to thank **Prof. Dr. Dhanya D, M.E, Ph.D.**, Head of the School, for his continuous support and **Mrs. Janila J, M.E**, for guidance and providing all technical support and for her support during the course of work.

Finally, we offer our sincere thanks to our parents, family members and all our friends for their valuable suggestions, encouragements and support throughout the entire project work.

TABLE OF CONTENT

CHAPTER No.	TITLE	PAGE No.
1	INTRODUCTION	6
1.1	Project Overview	6
1.2	Purpose	
2	LITERATURE SURVEY	7
3	IDEATION & PROPOSED SOLUTION	
3.1	Empathy Map Canvas	9
3.2	Ideation & Brainstorming	10
3.3	Proposed Solution	14
3.4	Problem Solution fit	15
4	REQUIREMENT ANALYSIS	
4.1	Functional requirement	16
4.2	Non-Functional requirements	17
5	PROJECT DESIGN	
5.1	Data Flow Diagrams	18
5.2	Solution & Technical Architecture	19
5.3	User Stories	20
6	PROJECT PLANNING & SCHEDULING	
6.1	Sprint Planning & Estimation	22
6.2	Sprint Delivery Schedule	23
6.3	Reports from JIRA	23
7	CODING & SOLUTIONING	
7.1	Feature 1	24
7.2	Feature 2	26
8	TESTING	
8.1	Test Cases	27
8.2	User Acceptance Testing	28
9	RESULTS	
9.1	Performance Metrics	29
10	ADVANTAGES & DISADVANTAGES	
10.1	Advantages	30
10.2	Disadvantages	30
11	CONCLUSION	31
12	FUTURE SCOPE	32

13	APPENDIX	
	Source Code	33
	GitHub & Project Demo Link	

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

News is one of the primary source of gaining information about the actions and events that happen all around. It may be an event that happened in the past, happening now or going to happen in the future. In the present days where there is a rapid increase in the development and adaptability of technologies throughout all the demographic of people, it is necessary to provide news in such a way that it is interconnected with the current technological trends. 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 is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds and so on as on official websites.

1.2 PURPOSE

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

CHAPTER 2

LITERATURE SURVEY

S.No	PAPER TITLE	AUTHOR	SOURCE	METHODS	RESOURCE LINK
1	Exploring mobile news reading interactions for news app personalization	Marios Constantinides, John Dowell, David Johson, Sylvain Malacria	ResearchGate	1. Identification of news reader types 2. Interaction logging and classification study 3. Deployment and data collection 4. Predicting News readertypes 5. Adaptive UI	https://www.researchgate.net/publication/299870645_Exploring_mobile_news_reading_interactions_for_news_app_personalisation
2	Detection and Tracking in News Articles	Sagar Patel, Sanket Suthar, Sandip Patel, Neha Patel	ResearchGate	1. Preprocessing 2. Tokenization 3. Stemming/Lemmization 4. Vector SpaceModel 5. Topic tracking	https://www.researchgate.net/publication/315657099_Topic_Detection_and_Tracking_in_News_Articles
3	Following the Fed with a News Tracker	Michael William McCracken	ResearchGate	1. The paper is a statistical paper 2. Data have come in stronger, weaker or as expected. 3. CitiGroup U.S Economic Surprise Index.	https://www.researchgate.net/publication/227438253_Following_the_Fed_with_a_News_Tracker
4	An End-to-end Weaklysupervised News Aggregation Framework	Xijin Tang, Xiaohui Huang	ResearchGate	1. Snorkelbased weaklysupervised classification 2. Latent Dirichlet Allocation (LDA) topic modeling, and topic signal detection 3. the temporal topic relationships and get the visualized results of news aggregation.	https://www.researchgate.net/publication/361087328_An_End-to-end_Weakly-supervised_News_Aggregation_Framework
5	An Approach to News Event Detection and Tracking Based on Stream of Online News	Yajie Qi, Li Zhou, Huayou Si, Jian Wan, Ting Jin.	IEEE Xplore	1. Event detection and tracking technology 2. Based on the idea of single-pass clustering algorithm. 3. Algorithm has a good effect on hot event detection.	https://ieeexplore.ieee.org/document/8048142

6	Android News App	Brijesh Joshi, Nehal Patel.	Research India Publications	<ol style="list-style-type: none"> 1. Fast and information oriented world we need to stay updated with every incidents and news too. 2. Application is to connect news articles from all around the world and deliver it to user as fast as possible in best visualize way. 	https://www.ripublication.com/ijaer18/ijaerv13n11_78.pdf
7	Research on Topic Detection and Tracking for Online News Texts	Guixian Xu, Yueting Meng, Zhan Chen, Xiaoyu Qiu, Changzhi Wang, Haishen Yao.	IEEE Xplore	<ol style="list-style-type: none"> 1. Topic tracking 2. LDA model is used to extract topics from news texts 3. Gibbs Sampling method is used to speculate parameters. 4. single-pass algorithm is used to track news topics. 5. The JS (Jensen-Shannon) divergence 	https://ieeexplore.ieee.org/document/8703401

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP

An empathy map is a simple, easy-to-digest visual that captures knowledge about user's behaviours and attitudes. It is a useful tool to help teams better understand their users.

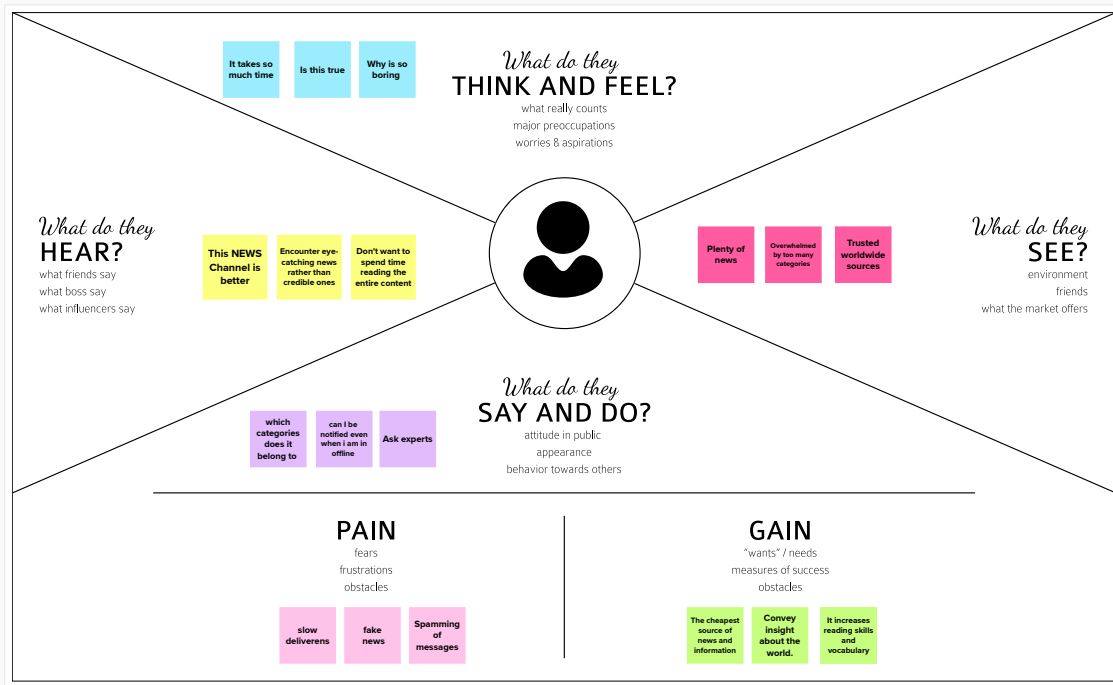
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1


Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 IDEATION & BRAINSTROMING

While the user signs up, ask for which categories they'll like to view the news on. And only push notifications regarding those categories. Also, use this category to show relevant news. Show 8 out of 10 new related to the selected category only. And, an option to edit the preferences. Ask the user to set up Do Not Disturb Time. During this time, do not push any notifications to the user. Providing customizable options to the user to select the time and frequency of notifications required. Allowing the user to customize the topics for which he requires to receive notifications By continuously taking feedback from the users regarding the notifications are they creating any disturbance or showing the relevant content for the customer?

Template




Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 2-8 people recommended

[Share template feedback](#)



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#)

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

8 minutes


PROBLEM

fake news
receiving time is too late
notification sending through SMS
NEWS can read through their Mother Tongue

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.



Need some inspiration?

Use a filtered version of this template to bootstrap your work.

[Open example](#)

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Notification of Important NEWS through SMS	Recording option	Speed reading option	Later reading option with full clarity
Enable of Audio option	Enable of Audio option	Notification of Rules and Scheme	NEWS can read through their Mother Tongue

Person 2

Finding of genuine NEWS	Interactive	Overcome copyright Issues
App size small	Usage of Data storage Low	Multiple usage

Person 3

Visible font	Negate irrelevant News	Reduce spammed notification
Download option	Deliverance of News at right time	personal preference

Person 4

NEWS Privilege	Catching NEWS rather than create one	Topic specification
Deliverance of NEWS at right time	Locational reading	Preferable topic specification



3

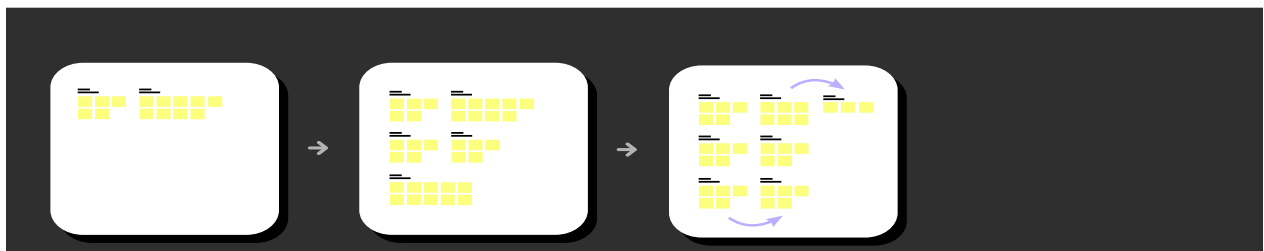
Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

**TIP**

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes

**After you collaborate**

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

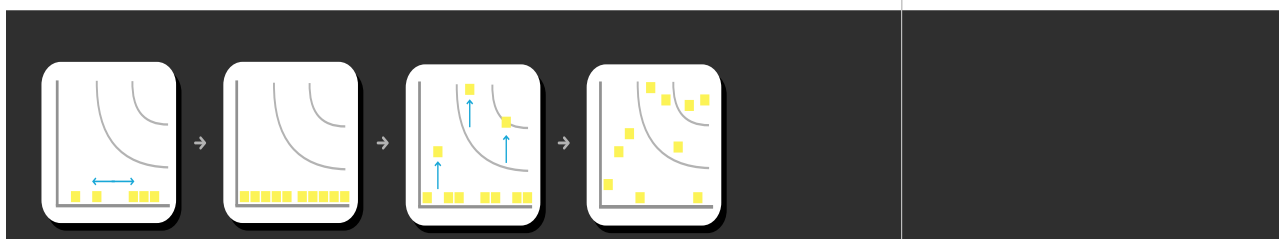
Quick add-ons

- A Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- B Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

- Strategy blueprint**
Define the components of a new idea or strategy.
[Open the template →](#)
- Customer experience journey map**
Understand customer needs, motivations, and obstacles for an experience.
[Open the template →](#)
- Strengths, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template →](#)

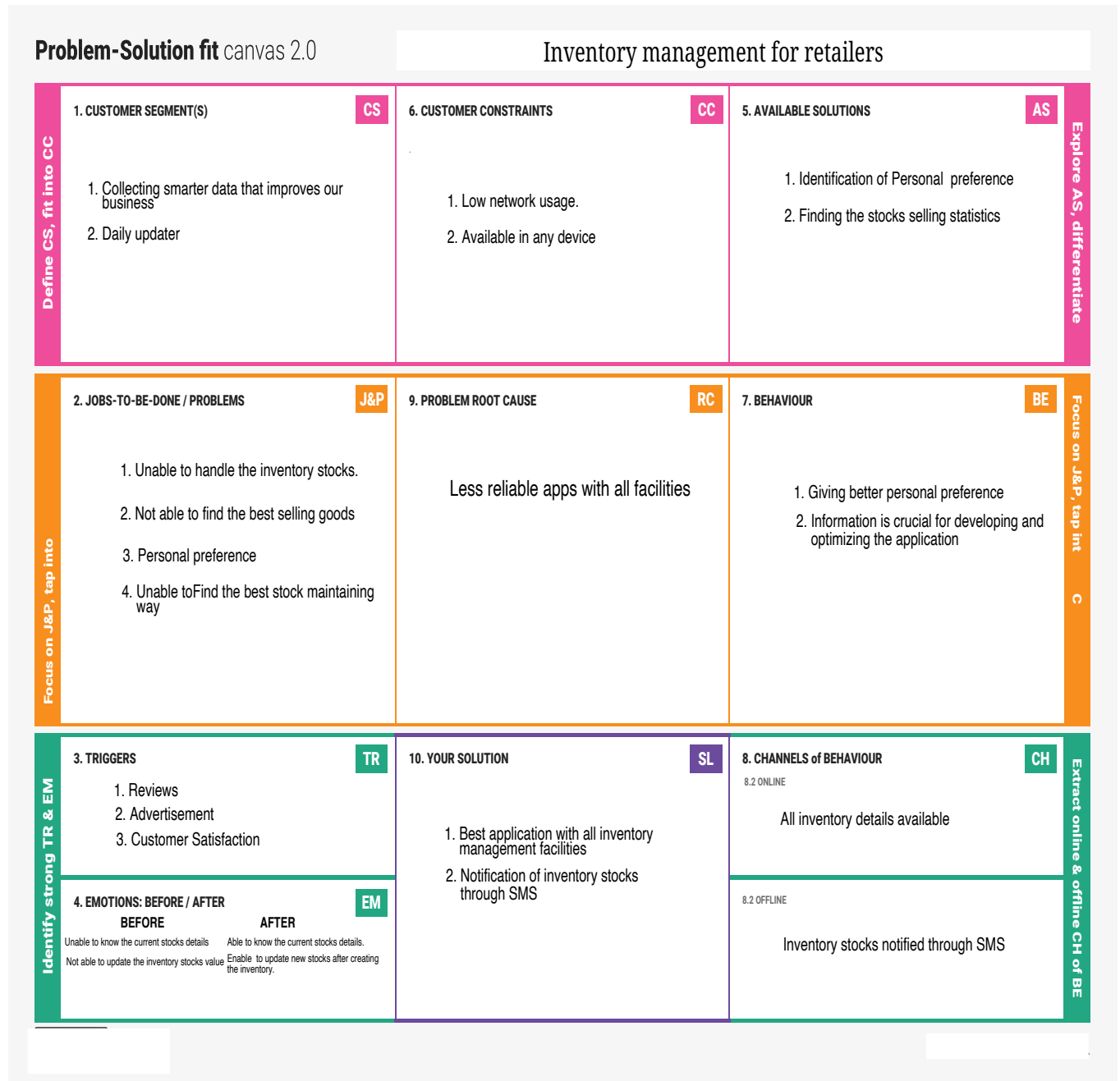
[Share template feedback](#)



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description																																				
1.	Problem Statement (Problem to be solved)	1. Unable to know the NEWS without internet 2. Not able to hear NEWS when working 3. Identification Personal preference																																				
2.	Idea / Solution description	1. Notification of Important NEWS through SMS 2. Enable Audio option 3. Using ML algorithm																																				
3.	Novelty / Uniqueness	1. SMS notification 2. ML algorithm																																				
4.	Social Impact / Customer Satisfaction	1. The world to your doorstep 2. Accessible in Various Language 3. Cheapest in per capita viewing 4. Upgrades Vocabulary and Grammar 5. Simple to Access and Read																																				
5.	Business Model (Revenue Model)	<table><tr><th></th><th>Empirical findings</th><th>International Newspaper</th><th>The Interpress</th><th>Community Micronews</th><th>Local Newspaper</th></tr><tr><td>Customers</td><td>Nicemarketing, readers Nichemarketing, advertisers Customer reading habits</td><td>Global, brand-aware readers Global advertisers "Global readers" Anytime, anywhere</td><td>Global readers Global and local advertisers Source awareness Anytime, anywhere</td><td>Community readers Community advertisers Local awareness Traditional, routine readers</td><td>Regional readers Regional advertisers Dedicated readers</td></tr><tr><td>Competition</td><td>Content Time Distribution</td><td>Other int. media houses 24/7 Distribution ownership</td><td>Local + global newspapers 24/7 Distribution network</td><td>Morning Distribution network</td><td>Other local newspapers Regularity Distribution ownership</td></tr><tr><td>Offering</td><td>Price model Device features Content features</td><td>Price premium Prestigious devices Uptodate worldnews</td><td>Reader price pressure Advertising price premium News wholesale Standard device Uptodate worldnews</td><td>Low price Standard device Extremely local news and ads</td><td>Subscription rate Branded device Local news and ads</td></tr><tr><td>Activities and Organization</td><td>Distribution Editorial workflow Advertising workflow New activities</td><td>Supply chain mgmt 24/7 Global advertisers</td><td>Provider relations 24/7 Differentiated advertising Umbrella branding Legal functions</td><td>Local provider relations One edition Micro advertising New mediator roles Umbrella branding</td><td>Owner of distribution Several editions Local and national advertisers New mediator roles</td></tr><tr><td>Resources / Factor market interaction</td><td>Journalism Advertising skills Perceived credibility</td><td>Well recognized journalists Global brand name Additional channel Serious journalism</td><td>Source reach Global and local brands Advertising networks Recognized wholesaler</td><td>Community journalists Multimedia journalists Local brands Local relations, small firms Authenticity</td><td>Multimedia journalists Local and national advertisers Additional channel Established brand</td></tr></table>		Empirical findings	International Newspaper	The Interpress	Community Micronews	Local Newspaper	Customers	Nicemarketing, readers Nichemarketing, advertisers Customer reading habits	Global, brand-aware readers Global advertisers "Global readers" Anytime, anywhere	Global readers Global and local advertisers Source awareness Anytime, anywhere	Community readers Community advertisers Local awareness Traditional, routine readers	Regional readers Regional advertisers Dedicated readers	Competition	Content Time Distribution	Other int. media houses 24/7 Distribution ownership	Local + global newspapers 24/7 Distribution network	Morning Distribution network	Other local newspapers Regularity Distribution ownership	Offering	Price model Device features Content features	Price premium Prestigious devices Uptodate worldnews	Reader price pressure Advertising price premium News wholesale Standard device Uptodate worldnews	Low price Standard device Extremely local news and ads	Subscription rate Branded device Local news and ads	Activities and Organization	Distribution Editorial workflow Advertising workflow New activities	Supply chain mgmt 24/7 Global advertisers	Provider relations 24/7 Differentiated advertising Umbrella branding Legal functions	Local provider relations One edition Micro advertising New mediator roles Umbrella branding	Owner of distribution Several editions Local and national advertisers New mediator roles	Resources / Factor market interaction	Journalism Advertising skills Perceived credibility	Well recognized journalists Global brand name Additional channel Serious journalism	Source reach Global and local brands Advertising networks Recognized wholesaler	Community journalists Multimedia journalists Local brands Local relations, small firms Authenticity	Multimedia journalists Local and national advertisers Additional channel Established brand
	Empirical findings	International Newspaper	The Interpress	Community Micronews	Local Newspaper																																	
Customers	Nicemarketing, readers Nichemarketing, advertisers Customer reading habits	Global, brand-aware readers Global advertisers "Global readers" Anytime, anywhere	Global readers Global and local advertisers Source awareness Anytime, anywhere	Community readers Community advertisers Local awareness Traditional, routine readers	Regional readers Regional advertisers Dedicated readers																																	
Competition	Content Time Distribution	Other int. media houses 24/7 Distribution ownership	Local + global newspapers 24/7 Distribution network	Morning Distribution network	Other local newspapers Regularity Distribution ownership																																	
Offering	Price model Device features Content features	Price premium Prestigious devices Uptodate worldnews	Reader price pressure Advertising price premium News wholesale Standard device Uptodate worldnews	Low price Standard device Extremely local news and ads	Subscription rate Branded device Local news and ads																																	
Activities and Organization	Distribution Editorial workflow Advertising workflow New activities	Supply chain mgmt 24/7 Global advertisers	Provider relations 24/7 Differentiated advertising Umbrella branding Legal functions	Local provider relations One edition Micro advertising New mediator roles Umbrella branding	Owner of distribution Several editions Local and national advertisers New mediator roles																																	
Resources / Factor market interaction	Journalism Advertising skills Perceived credibility	Well recognized journalists Global brand name Additional channel Serious journalism	Source reach Global and local brands Advertising networks Recognized wholesaler	Community journalists Multimedia journalists Local brands Local relations, small firms Authenticity	Multimedia journalists Local and national advertisers Additional channel Established brand																																	
6.	Scalability of the Solution	The website/app attendance limit must be scalable enough to support 200,000 users at a time.																																				

3.4 PROPOSED SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNTIONAL 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 Form Registration through Gmail Registration through LinkedIn Registration through Phone number
FR-2	User Confirmation	Confirmation via OTP(Email) Confirmation via OTP(SMS)
FR-3	Transactions processing	Payment through Online
FR-4	Authentication	Through Email Through SMS
FR-5	Reporting	Through App Through Email Through SMS

4.2 NON- FUNCTIONAL REQUIREMENT

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

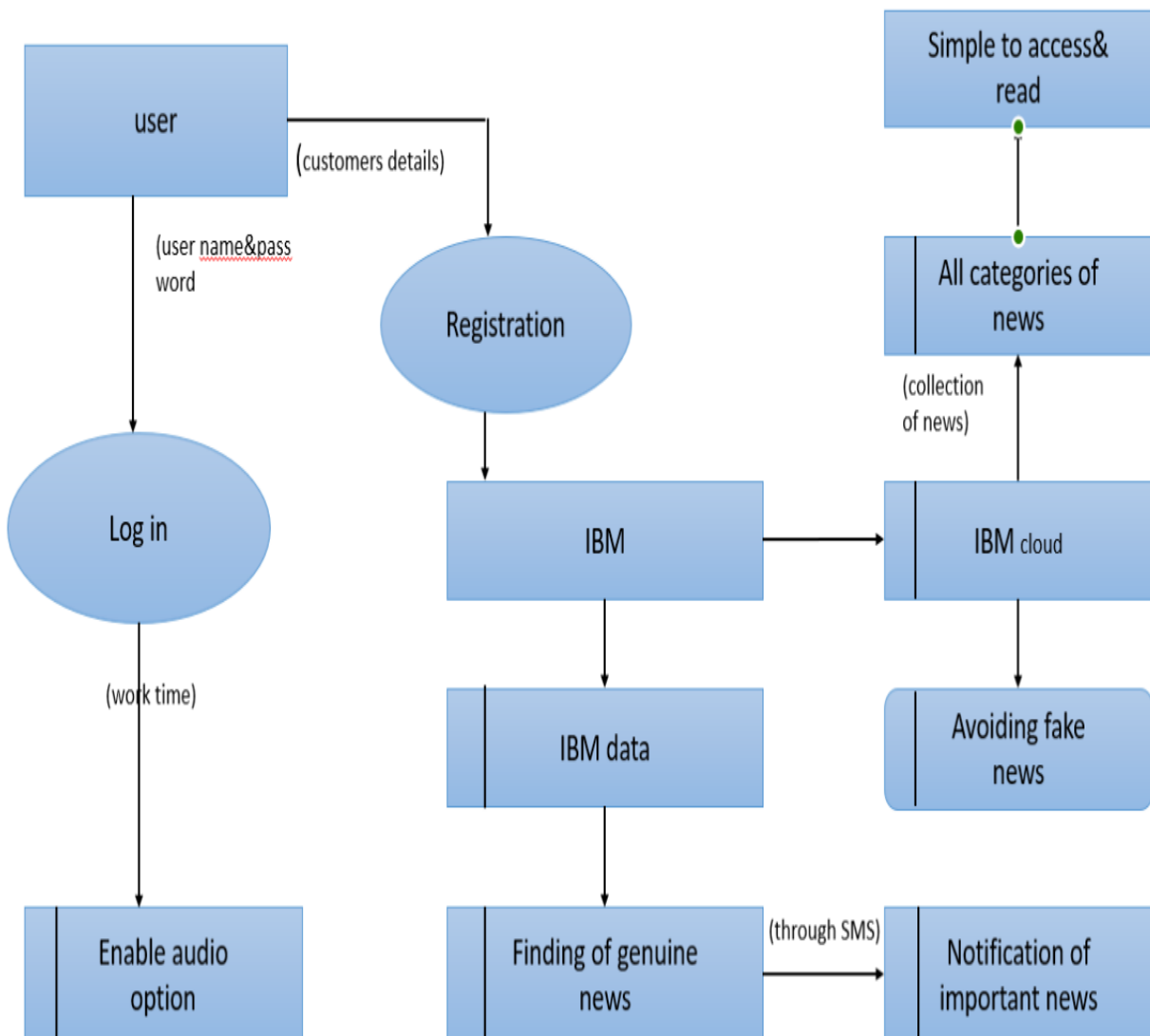
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	24/7
NFR-2	Security	Access permissions for the particular system information may only be changed by the system's data administrator.
NFR-3	Reliability	Automatic restart and operation recovery.
NFR-4	Performance	The front-page load time must be no more than 2 seconds for users that access the website using an LTE mobile connection.
NFR-5	Availability	New module deployment mustn't impact front page, product pages, and check out pages availability and mustn't take longer than one hour. The rest of the pages that may experience problems must display a notification with a timer showing when the system is going to be up again.
NFR-6	Scalability	The website/app attendance limit must be scalable enough to support 200,000 users at a time.

CHAPTER 5

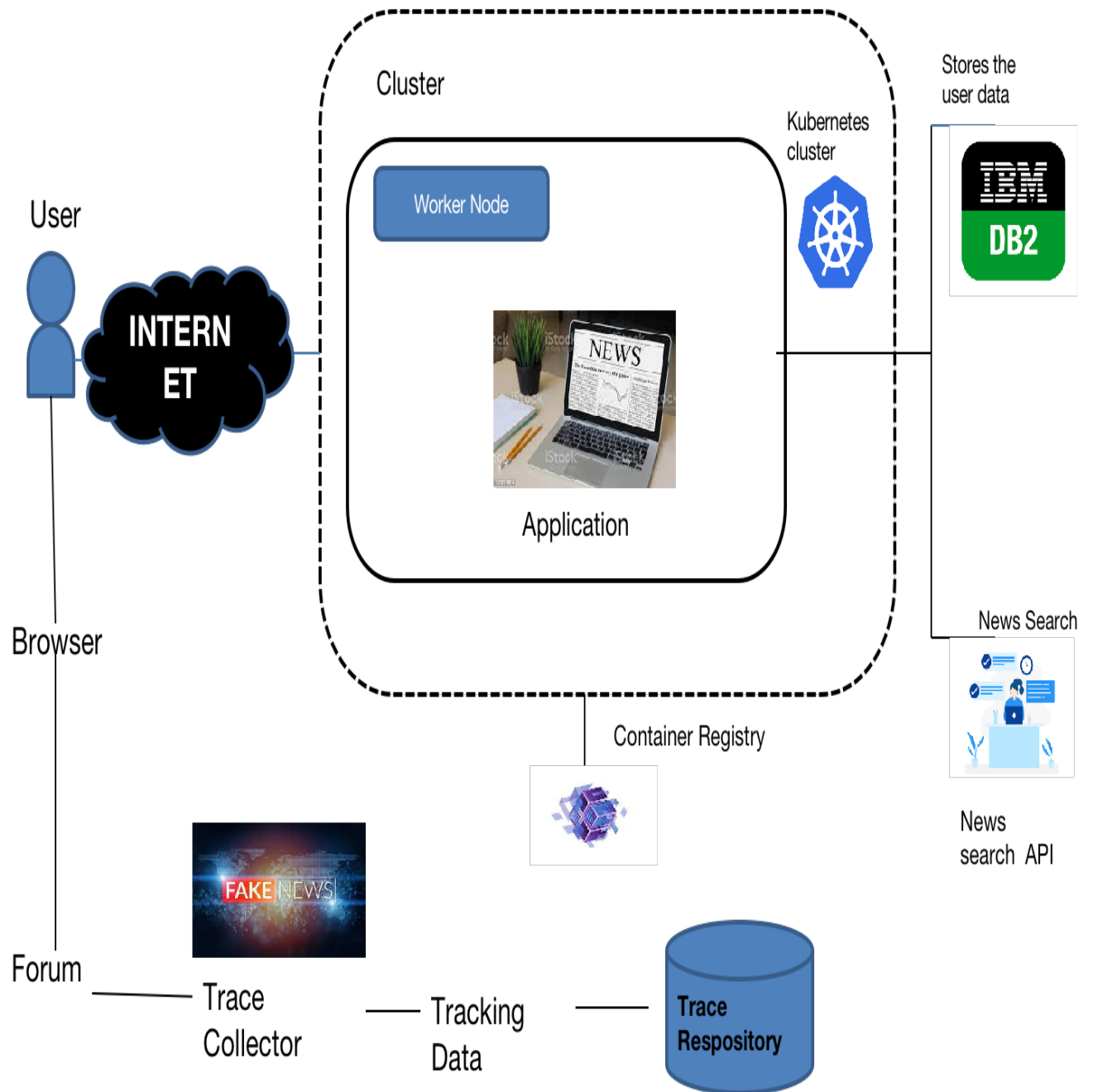
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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 & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story no	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 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can access my account	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation email and password through email	High	Sprint-1
	Dashboard	USN-6	As a user, I will able to see the login and profile details	I can able to login through Gmail account	Medium	Sprint-3
Customer (Web user)	Registration	USN-7	As a user, I can register	I can access my google	High	Sprint-1

			for the application by entering my name, phone number, email, password, and confirming my password.	account / dashboard		
Customer Care Executive	Chat box	USN-8	As a customer care executive, I can ask about the inconvenience and the usability about the app	I can access through IBM Watson	Medium	Sprint-4
Administrator	Login	USN-9	As a administrator, I can help to login and register the user through gmail	I can receive confirmation Gmail to user	High	Sprint-2
NEWS Tracker	IBM cloud	USN-10	As a tracker, I collect all the NEWS from the cloud	I can access the IBM Cloud	Medium	Sprint-1
Database Manager	IBM DB2	USN-11	As a database manager, I can collect all the user details and NEWS application details	I can access the IMB Db2	Medium	Sprint-3

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

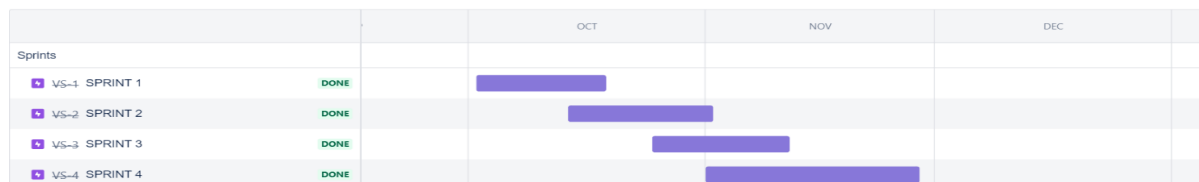
Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Ajisha Jegal Mersheba Fred
Sprint-1		USN-2	As a user, I can register for the application through Facebook	1	Medium	Kebiya Adlin Ancy
Sprint-1		USN-3	As a user, I can register for the application through Gmail	1	Low	Kebiya
Sprint-1	Confirmation	USN-4	As a user, I will receive confirmation email once I have registered for the application	2	Medium	Ajisha Jegal Kebiya
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Ajisha Jegal
Sprint-2	Dashboard	USN-6	As a user I should be able to navigate and access all the features hassle free	2	High	Mersheba Fred
Sprint-2	Layout	USN-7	As a user I should be able to access the portal with different devices with the same comfort	2	Medium	Adlin Ancy
Sprint-3	Data Store and retrieval	USN-8	Get Data from API and store as JSON in DB2	3	Medium	Kebiya
Sprint-3		USN-9	Get bin data from API and store in DFS	2	High	Ajisha Jegal Adlin Ancy
Sprint-4	User Segregation and data access	USN-10	As a CC executive I should be able to uniquely identify the customer and offer help	1	Low	Mersheba Fred Adlin Ancy
Sprint-4	Change code	USN-11	As a administrator I should be able to modify code according to the future requirements.	2	Medium	Kebiya Adlin Ancy
Sprint-4	Monitor the system	USN-12	As a administrator I should be able to monitor the cloud system and fix errors before customer	1	High	Ajisha Jegal Mersheba Fred

6.2 SPRINT DELIVERY SCHEDULE

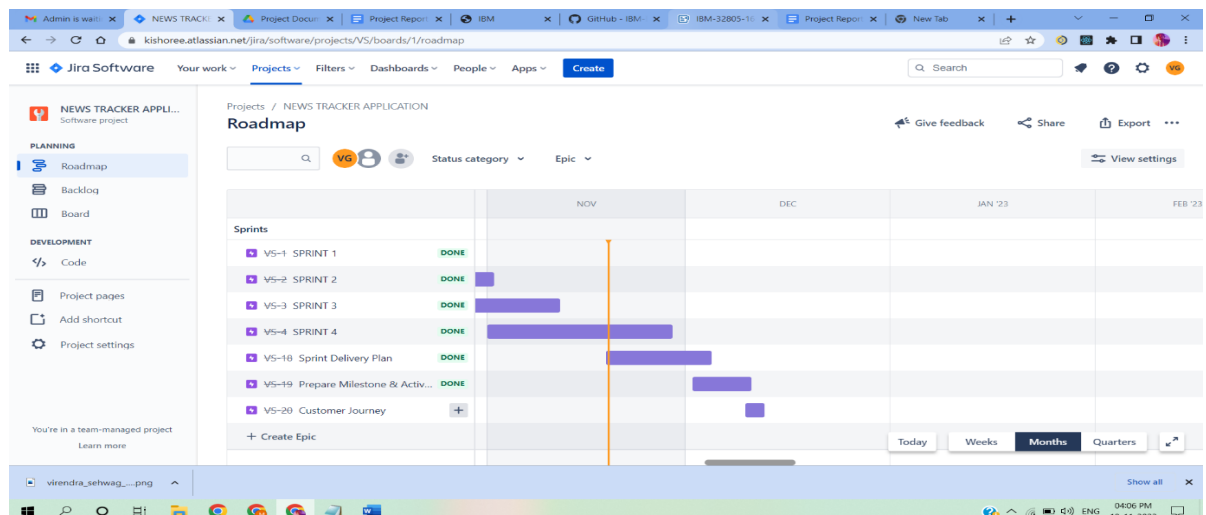
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (ason Planned End Date)	Sprint Release Date(Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	5	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	4	6 Days	14 Nov 2022	19 Nov 2022	4	19 Nov 2022

6.3 REPORT FROM JIRA

SPRINT DETAILS



SPRINT OVERALL DETAILS



CHAPTER 7

CODING AND SOLUTIONING

7.1 FEATURE 1

Voice Recognition:

We can search through the voice input also. So, that user can use their mobile device single handily. 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/ver
sions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
  document.head.appendChild(t);
});

```

CHAPTER 8

TESTING

8.1 TEST CASES

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.

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
Duplication	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

CHAPTER 9

RESULTS

9.1 PERFORMANCE MATRIXS

These application performance metrics, commonly known as key performance indicators (KPIs), are a quantitative measure of how effectively the organization achieves the business objectives. Capturing the right metrics will give you a comprehensive report and powerful insights into ways to improve your application.



CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

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

10.2 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.

CHAPTER 11

CONCLUSION

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.

CHAPTER 12

FUTURE SCOPE

FUTURE SCOPE

- We are in a process of developing an 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.

CHAPTER 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=3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECURITY
=
SSL;SSLServerCertificate=credentials/DigiCertGlobalRootCA.crt;UID=bgh36879;PWD=T
v
SzQQQ6Jhaaiwg6","")
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")
        userMobile = request.form.get("userMobile")
        picture = request.form.get("picture")

```

```

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(prepare_stmt, 1, userName)
        ibm_db.bind_param(prepare_stmt, 2, userEmail)
        ibm_db.bind_param(prepare_stmt, 3, userPassword)
        ibm_db.bind_param(prepare_stmt, 4, userMobile)
        ibm_db.bind_param(prepare_stmt, 5, picture)
        ibm_db.execute(prepare_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='')

try:
    sg=SendGridAPIClient('SG.29Td0tbNSkylif9SSPnQNA.4DBECk8ka8RmmYRE5OIsRKG
OR2QI2raRG3CLmdsVBVc')

```

```

        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 successfully..
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')

# weather
@app.route('/weather')
def weather():
    return render_template('weatherinfo/weatherpage.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_in
put_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_input
_s
earch_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,ret
urned_input_se
arch_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_input_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_input_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_search_value=search_value_name,result_count=result_count)
    else:
        return
        render_template('notfound.html',responseData=searchResponse,returned_inpu
        t_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-45214-1660728857>

Project Demo Link:

<https://drive.google.com/file/d/1ksWnJ5n9rVdMXq94yPDe9Ymbh9LhWbgi/view?usp=drivesdk>