1.Introduction

1.1 Project overview

As our lives are very busy these days, we often feel we need more than 24 hours. 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, etc.... as on official websites.

we will create a News Web Application using Flask and News API. The web page will display top headlines and a search bar where the user can enter a query, after processing the query, the webpage will display all relevant articles (up to a maximum of 100 headlines). We will create a simple user interface using HTML and bootstrap.

API key generation

In order to use the News API in our application, we need to generate a unique API key from https://newsapi.org/. Visit this website and create your free account, on successful registration and email verification you will get your API key on the screen. While registering you may need to choose a developer plan (choose according to your requirements). Save this API key somewhere so that it can be used further.

Get articles related to a search query

This will be related to a post request We are going to use the get_everything function with parameters like query sources, domains, language, sorting based on, and page_size. We will fetch all sources with the help of the get_sources function of news api and to fetch all domains we will use 'URL' property of each source. Here is the code which shows how we are going to handle it:

1.2 Purpose

Newspaper is one of the most popular and required assets of our daily lives. And, in today's hectic world, reading newspaper has become one of the traditional ways of reading the news. With the news being created every minute and relayed through TV, radio and internet, the updated news is already old by the next morning. And, that's why newspaper and magazine publishers are struggling to keep-up with the pace. Change is needed and publishers must embrace mobile.

Today, the publishing industry is facing such a threat when it comes to newspaper publishing and sales. So, magazine and newspaper lovers are moving towards reading news on mobiles and tablets. The revenue model of the online apps are quite simple and rewarding. They run ads and generate a good amount of money.

Here are some amazing features which news app offers:
User Interface
It is an old saying "first impression is the last impression". So the developer should make sure that the application should leave a mark on the users. This is where you need to focus on bringing interactive, visual and architectural designs as well. It means that the content should be distributed in the app such that the screen do not appear crowded with the content.

Filter content

The option of filtering the content based on different category should be incorporated in the app to provide the audience wide taste and sensibilities.

Easy offline access

When a user is not online due to some reason he/she should have to access to the internet. Whenever the user is online the news content is downloaded in the cache memory of the app, this is how a user can access to the content offline.

Social Media integration

This will help the users to share news on various platforms such as Twitter and Facebook. This will not only give an amazing user experience and also will also increase the views.

Premium content

If you wish, a separate section for the premium content can be added to the app that you can make available for your privileged customers.

If these features are kept in mind then you will definitely have an amazing news app which will help the users get real time updates.

At AppInventiv, we believe in quality of work and customer satisfaction, hence we offer a set of flexible pricing models to our customers. Our clients have the full liberty to select the best option as per their business requirements. Our renowned strategy & planning team works alongside our world class branding and digital design experts which has made us development company. We create beautiful and engaging designs, matched only by the intuitive and innovative user interfaces that those designs are based on.

2. Literature Survey

2.1 Existing Problem

An Approach to News Event Detection and Tracking Based on Stream of Online News

Once an event occurs, usually there are a large number of online news to be released. How to quickly and accurately detect the hot events from the huge amount of online news is the focus and hotspot. Event detection and tracking technology is as a key technology to solve this problem. In this paper, we propose an approach to detect hot events from the online news stream in a timely manner and track the hot events. Based on the idea of single-pass clustering algorithm, this approach addresses the weight of keywords and proposes a new method to calculate similarity among news to track event. Through the analysis of the experimental results, we can find that this algorithm has a good effect on hot event detection.

Exploring Mobile News Reading Interactions for News App Personalisation

About the Problem:

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

Android News App

About the Problem:

As world’s technology is rapidly growing, we have fast connection and network to instantly connect to other person. Day to day use in mobile, tablets and laptop is increasing, most of the people already have this facilities. In this fast and information oriented world we need to stay updated with every incidents and news too. This News app is android mobile application where user have access to latest news from 120+ newspapers from 50+ countries. The main focus of this application is to connect news articles from all around the world and deliver it to user as fast as possible in best visualize way.

Research on Topic Detection and Tracking for Online News Texts

About the Problem:

With the rapid development of the Internet, the amount of data has grown exponentially. On the one hand, the accumulation of big data provides the basic support for artificial intelligence. On the other hand, in the face of such huge data information, how to extract the knowledge of interest from it has become a matter of general concern. Topic tracking can help people to explore the process of topic development from the huge and complex network texts information. By effectively organizing large-scale news documents, a method for the evolution of news topics over time is proposed in this paper to realize the tracking and evolution of topics in the news text set. First, the LDA model is used to extract topics from news texts and the Gibbs Sampling method is used to speculate parameters. The topic mining using the K-means method is compared to highlight the advantages of using LDA for topic discovery.

Second, the improved single- pass algorithm is used to track news topics. The JS divergence is used to measure the topic similarity, and the time decay function is introduced to improve the similarity between topics with the similar time. Finally, the strength of the news topic and the content change of the topic in different time windows are analysed. The experiments show that the proposed method can effectively detect and track the topic and clearly reflect the trend of topic evolution.

2.2 Reference

Source: ResearchGate

Authors: Marios Constantinides, John Dowell, David Johnson, Sylvain Malacria.

Websites:
https://www.researchgate.net/publication/299870645_Exploring_mobile_news_reading_interactions_for_news_app_personalisation
Source: Research India Publications

Authors: Brijesh Joshi, Nehal Patel.
Websites: https://www.ripublication.com/ijaer18/ijaerv13n11_78.pdf
Source: IEEE Xplore

Authors: Guixian Xu, Yueting Meng, Zhan Chen, Xiaoyu Qiu, Changzhi Wang, HaishenYao.
Websites: https://ieeexplore.ieee.org/document/8703401

2.3 Problem Statement Definition

There are multiple news-sharing apps used by a single user and are often spammed with notifications. There is also a lot of fake news which gets shared. A news-sharing app wants to help users find relevant and important news easily every day and also understand explicitly that the news is not fake but from proper sources.

Reasons for choosing

As I felt that there is a considerable hike in the news reading population due to COVID 19 and Work at home. My team members found it acceptable. Due to changes brought about by the pandemic, people started depending on the news a lot to get updates about the situation around them. We thought that this behaviour change would be very interesting to study. Based on my suggestion my team made it a criterion.

None of us had previously seen a UX case study done on this topic and we felt like this is a good industry to solve for since it caters to a wide range of people (culturally, economically, and socially). We also felt that news apps are used by a lot of people but it feels like a space that hasn’t been explored much and has a lot of potential scope for betterment.

As a team, we realized that we have different skill sets and felt that we would all be able to cater to this problem statement in a very strategic and collaborative manner. Plus it was interesting to see how all of us had a very different take on news consumption.

I have a strong background in multimedia and animation which has a close connection with news and communication. I would be able to contribute additionally to the content through the User Interface designing part and

Video Editing also.

Haresh is good at analysing research data and hence would be able to contribute to defining the problems and understanding the psyche of the users. and manjunath is very good at conducting desk research. abijith nikash and nandha kumar would render cooperation in each of their sections.

Assumptions about the User

Spamming of messages usually leads to clearing of the content without viewing thus probably leading the user to lose access to important information.

Too many notifications irritate the users and it clutters the bar on their phone.

Irrelevant news makes the user stop viewing the news thus losing access to credible news.

Users don't want to spend time reading the entire content. They need short and crisp news.

Using multiple apps because users are not getting the content categories they want in one app.

The user doesn't want to waste time figuring out the relevance of the news
Young people don't read news from apps, usually depend on social media to get updated.

Users sometimes get overwhelmed by too many categories.

Users only check their phones and go through these notifications when they are free or during a particular time of the day when they are doing time pass Ads in the apps might irritate the user while reading the news.

One app not accommodating both the regional and international news might create a requirement for different apps.

Assumptions about the business

News apps are trying to be like social media apps

Businesses have to publish irrelevant news because younger generations prefer news with more fun instead of reliable news.

News apps want to increase the time that user spends on their app so that they can show ads and generate revenue.

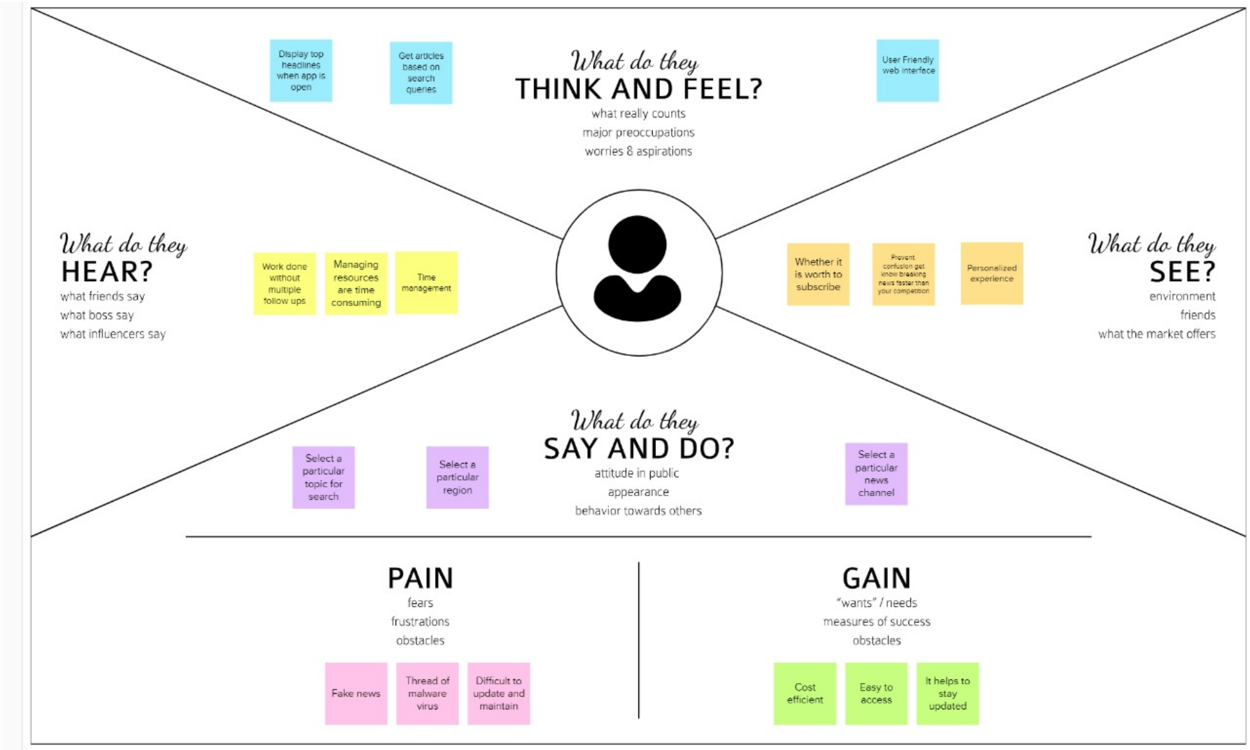
To increase the user screen time, news apps make users encounter eye-catching news rather than credible ones.

Assuming that this business started booming more during the pandemic

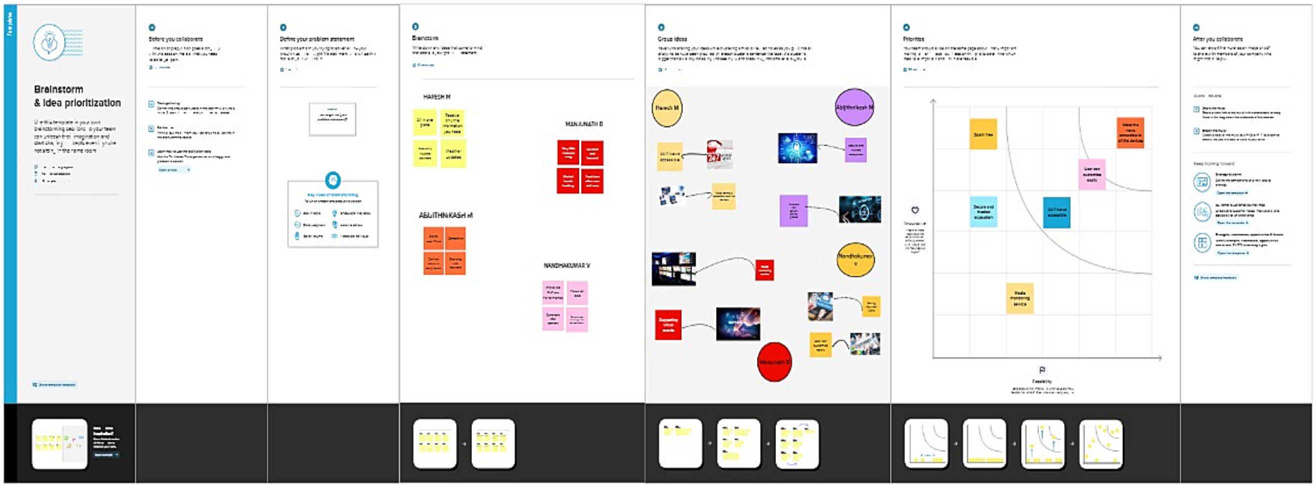
Apps generate income through subscriptions.

3. Ideation And Proposition

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

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 only in 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 their area of interest. So that they will be notified, if any new news is updated in their interested areas.

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

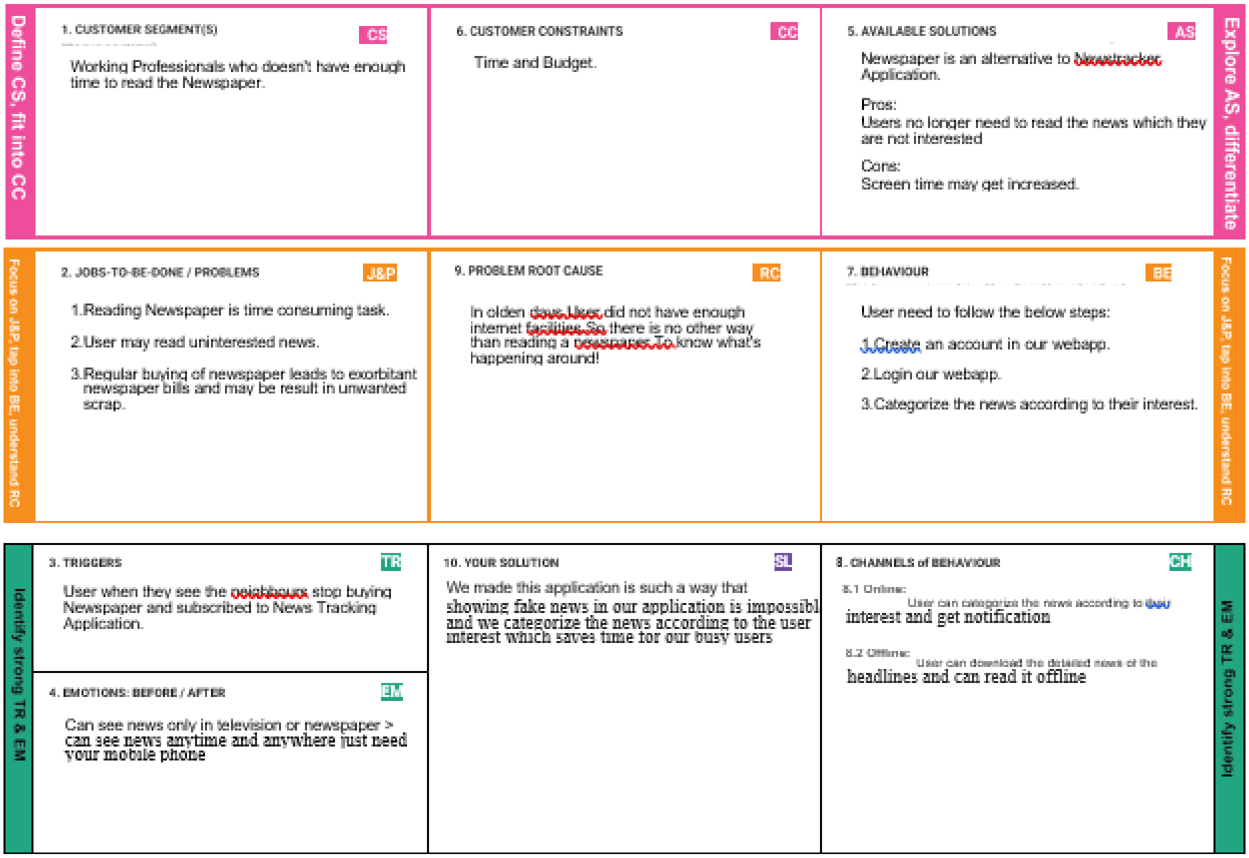
This application encourages its users to provide feedback. Based on that feedback, developments were made eventually.

5. Business Model (Revenue Model)

1. Add advertisements to the application, so that we can get revenue from those advertisement-sponsored organizations. More advertisements may irritate the user.

2. Add premium subscription, users who subscribe for premium won't get advertisements.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Working Professionals who doesn't have enough time to read the Newspaper.

6. CUSTOMER CONSTRAINTS CC

Time and Budget.

5. AVAILABLE SOLUTIONS AS

Newspaper is an alternative to Newstracker Application.

Pros:
Users no longer need to read the news which they are not interested

Cons:
Screen time may get increased.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

1.Reading Newspaper is time consuming task.

2.User may read uninterested news.

3.Regular buying of newspaper leads to exorbitant newspaper bills and may be result in unwanted scrap.

9. PROBLEM ROOT CAUSE RC

In olden days.User did not have enough internet facilities.So there is no other way than reading a newspaper.To know what's happening around!

7. BEHAVIOUR BE

User need to follow the below steps:

1.Create an account in our webapp.

2.Login our webapp.

3.Categorize the news according to their interest.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

User when they see the neighbours stop buying Newspaper and subscribed to News Tracking Application.

4. EMOTIONS: BEFORE / AFTER EM

Can see news only in television or newspaper > can see news anytime and anywhere just need your mobile phone

10. YOUR SOLUTION SL

We made this application is such a way that showing fake news in our application is impossibl and we categorize the news according to the user interest which saves time for our busy users

8. CHANNELS of BEHAVIOUR CH

8.1 Online:
User can categorize the news according to their interest and get notification

8.2 Offline:
User can download the detailed news of the headlines and can read it offline

Identify strong TR & EM

4. Requirement Analysis

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

User Registration	Registration through online application Registration throughGmail Registration through website
User Confirmation	Confirmation via Email Confirmation via OTP
User login	Login through browser directly by entering username and password Login through Login through email
User interaction	Done through user interface between client and server View the related news by subscripted or requested page

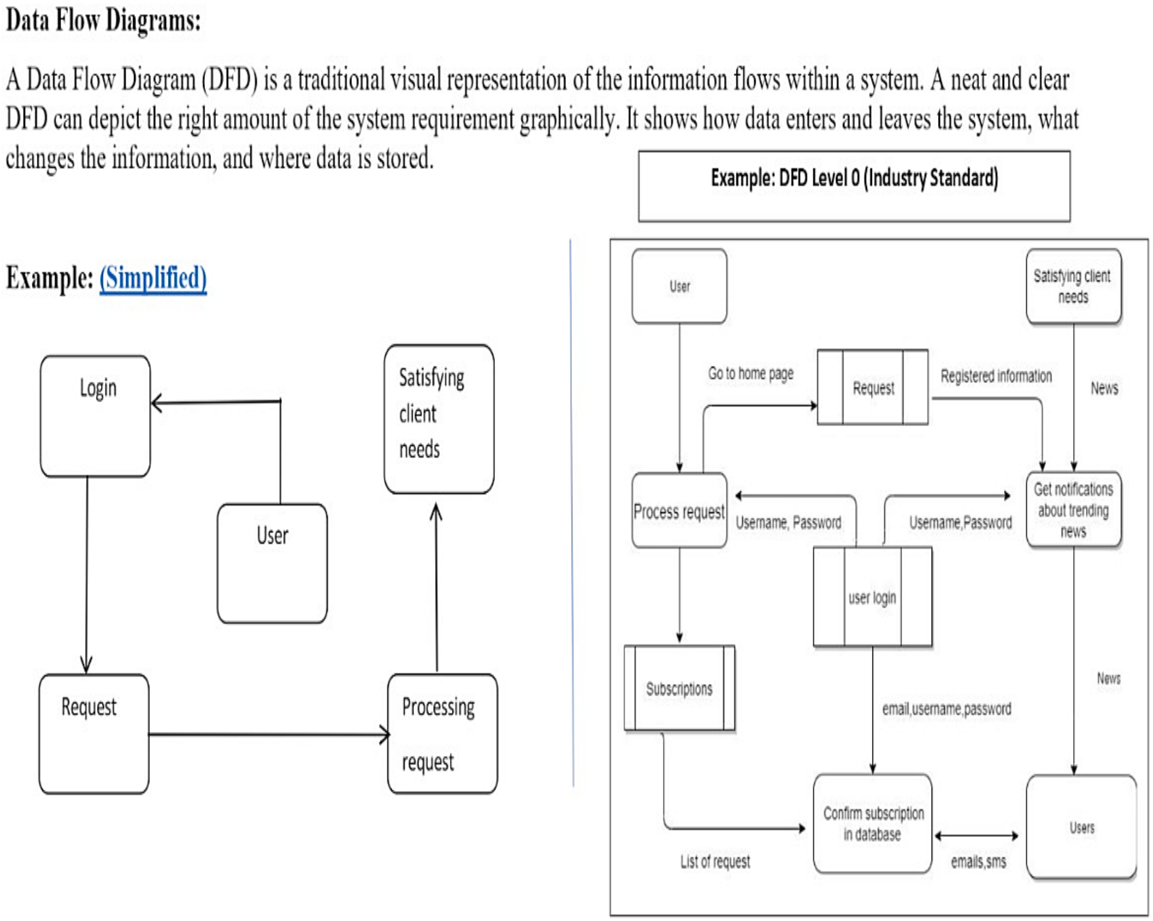
4.2 Non-Functional Requirements

Usability	End users can receive push updates for new content on a site by subscribing to the site’s news feed
Security	How well are the system and its data protected against attacks
Reliability	How often does the system experience critical failures? How much time does it take to fix the issue when it arises ?And how is user availability time compared to downtime?
Performance	Performance is the core non-functional requirements no system can do without.It defines how fast a software system or a particular piece of it responds to certain users actions under a certain workload. In most cases, this metric explains how long a user must wait before the target happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it’s not always like that. Performance requirements may describe background processes invisible to users, e.g. backup. But let’s focus on performance.

Availability	Availability describes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period. For instance, the system may be available 98 percent of the time during a month. Availability is perhaps the most business-critical requirement, but to define it, you also must have estimations for reliability and maintainability.
Scalability	Scalability assesses the highest workloads under which the system will still meet the performance requirements. There are two ways to enable your system scale as the workloads get higher: horizontal and vertical scaling.

5. Project Design

5.1 Data Flow Diagram



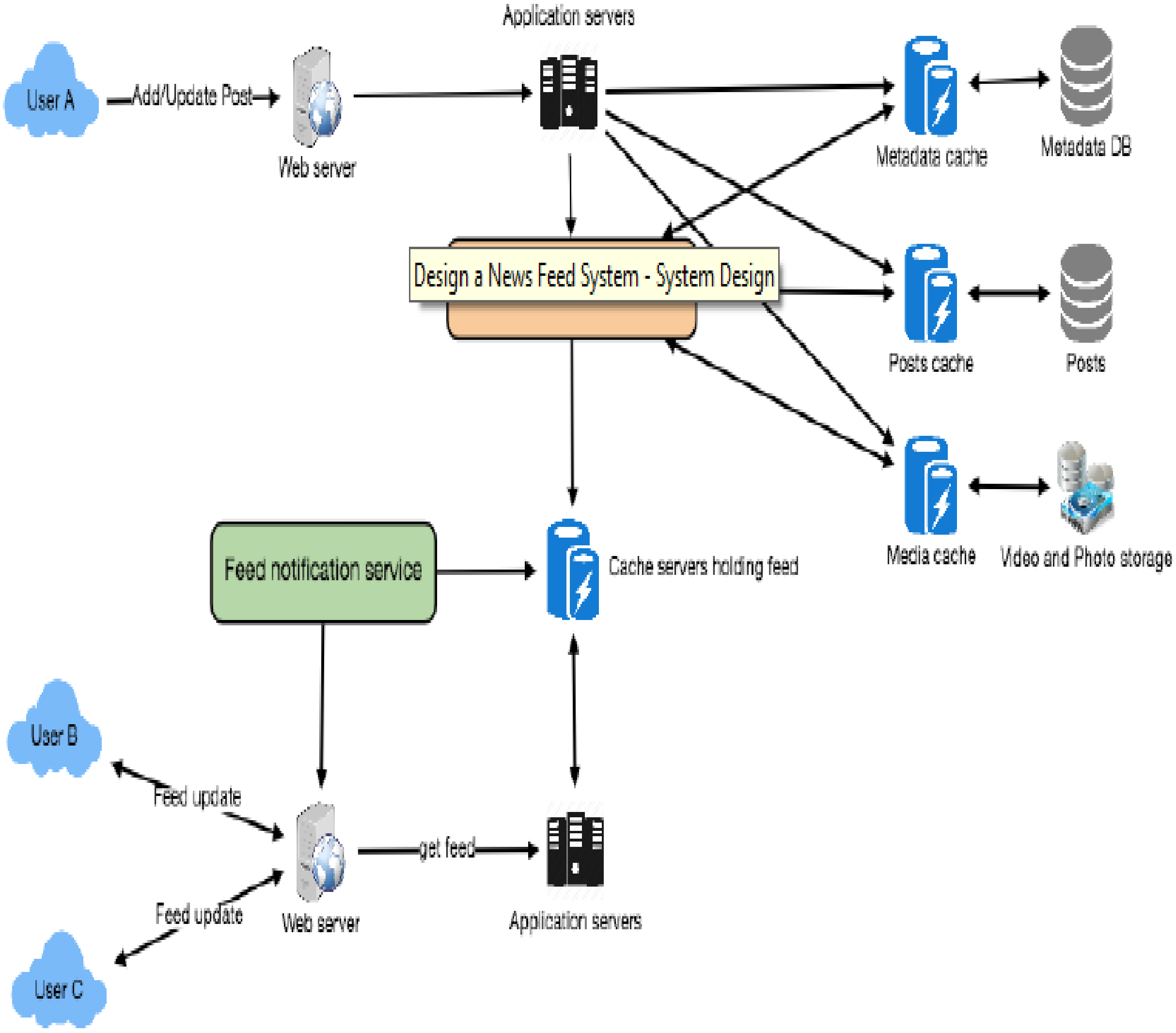
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

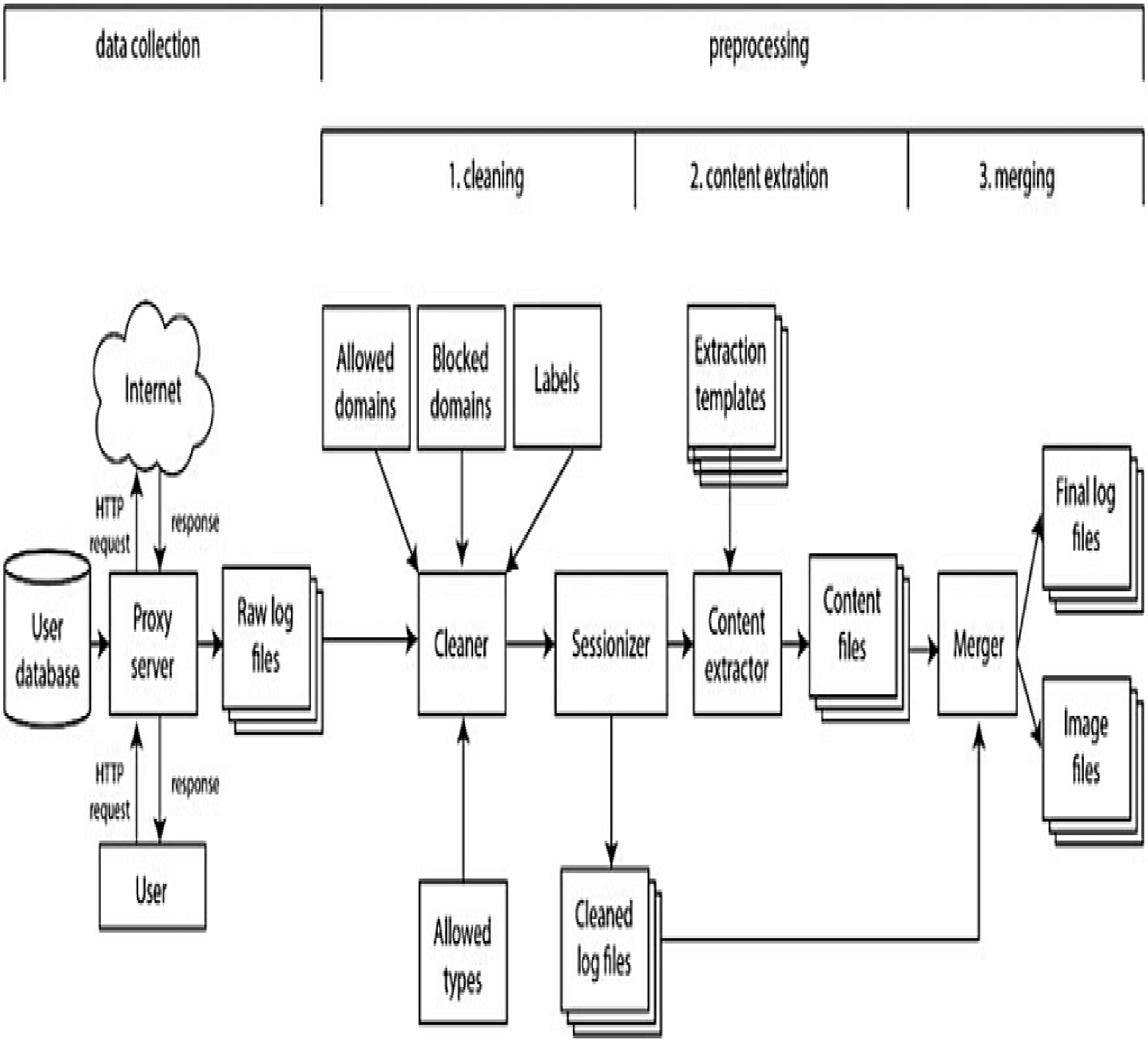
Example: (Simplified)

Example: DFD Level 0 (Industry Standard)

5.2 Solution & Technical Architecture



Technology Architecture



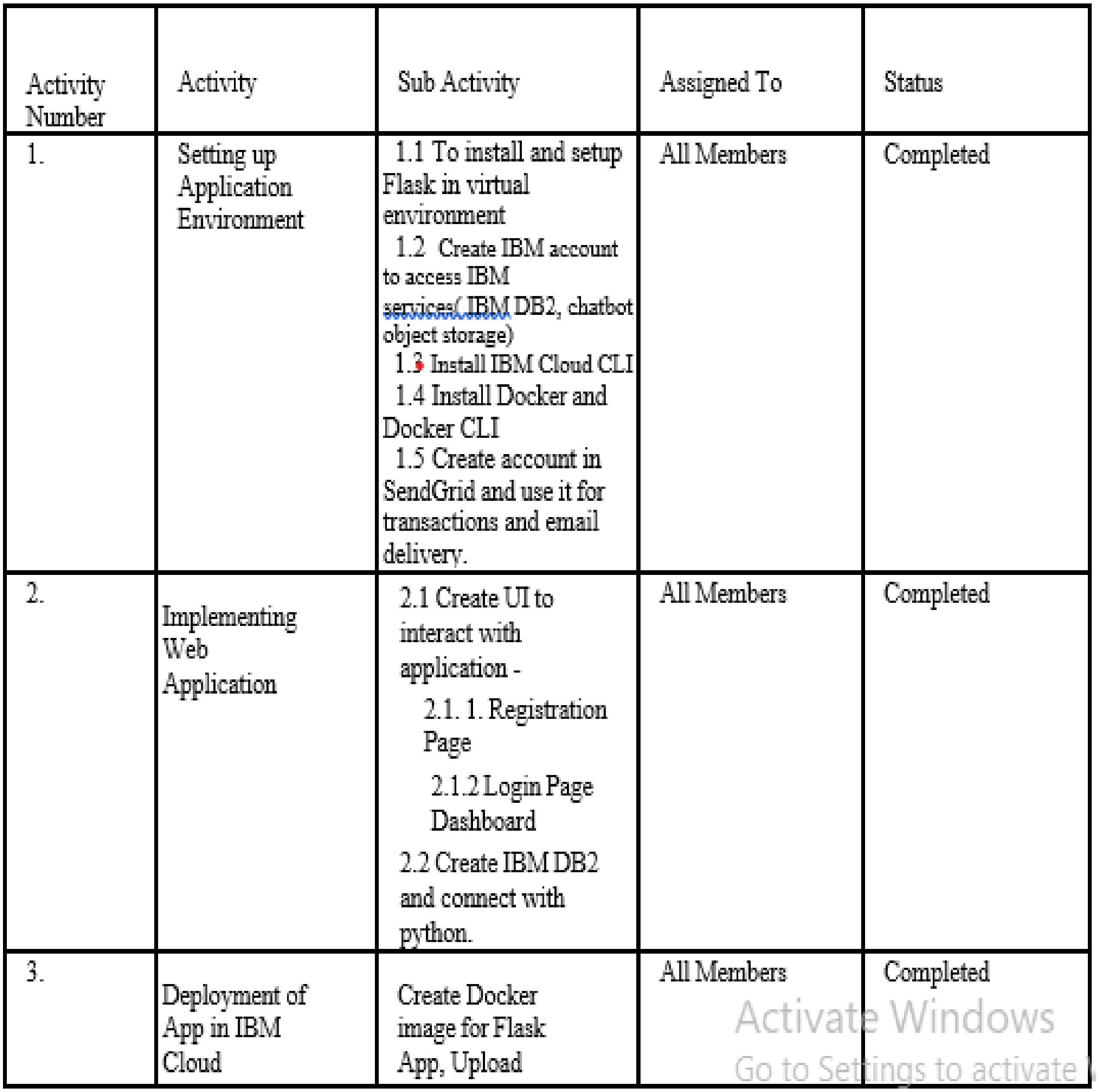
5.3 User Stories

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 (Searchi ngnews)	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 their given website	I can register & access the dashboard with Gmail or in Browser Login	Low	Sprint-2
		USN-4	As a user, I can register for theapplication 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 informations through this application	High	Sprint-1
	Dashboard	USN-6	To see their histories about recentlyviewed, updates for search related news, current progress ,feedback			
Customer (Web user)	Browser	USN-7	Have interactive medium betweenclient and server	I have a clarity to use this applicationand easily resolve my specific issues	high	Sprint-1
Customer Care Executive	Chat bot	USN-8	Rectify the is user related subscription,account,terms and conditions,privacy policy			

6. Project Planning & Scheduling

6.1 Sprint Planning And Estimation



Activity Number	Activity	Sub Activity	Assigned To	Status
1.	Setting up Application Environment	1.1 To install and setup Flask in virtual environment 1.2 Create IBM account to access IBM services(IBM DB2, chatbot object storage) 1.3 Install IBM Cloud CLI 1.4 Install Docker and Docker CLI 1.5 Create account in SendGrid and use it for transactions and email delivery.	All Members	Completed
2.	Implementing Web Application	2.1 Create UI to interact with application - 2.1. 1. Registration Page 2.1.2 Login Page Dashboard 2.2 Create IBM DB2 and connect with python.	All Members	Completed
3.	Deployment of App in IBM Cloud	Create Docker image for Flask App, Upload	All Members	Completed

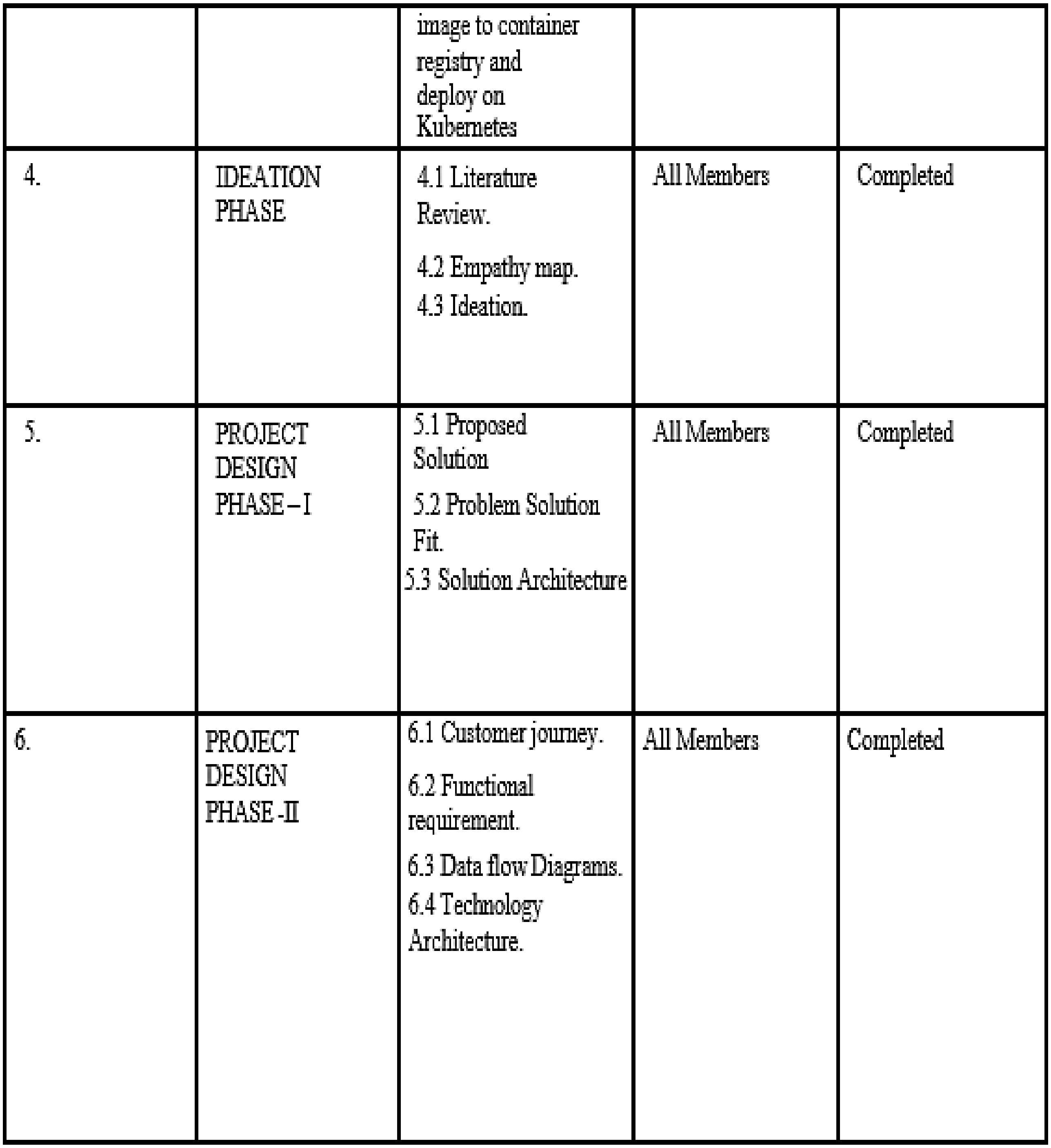
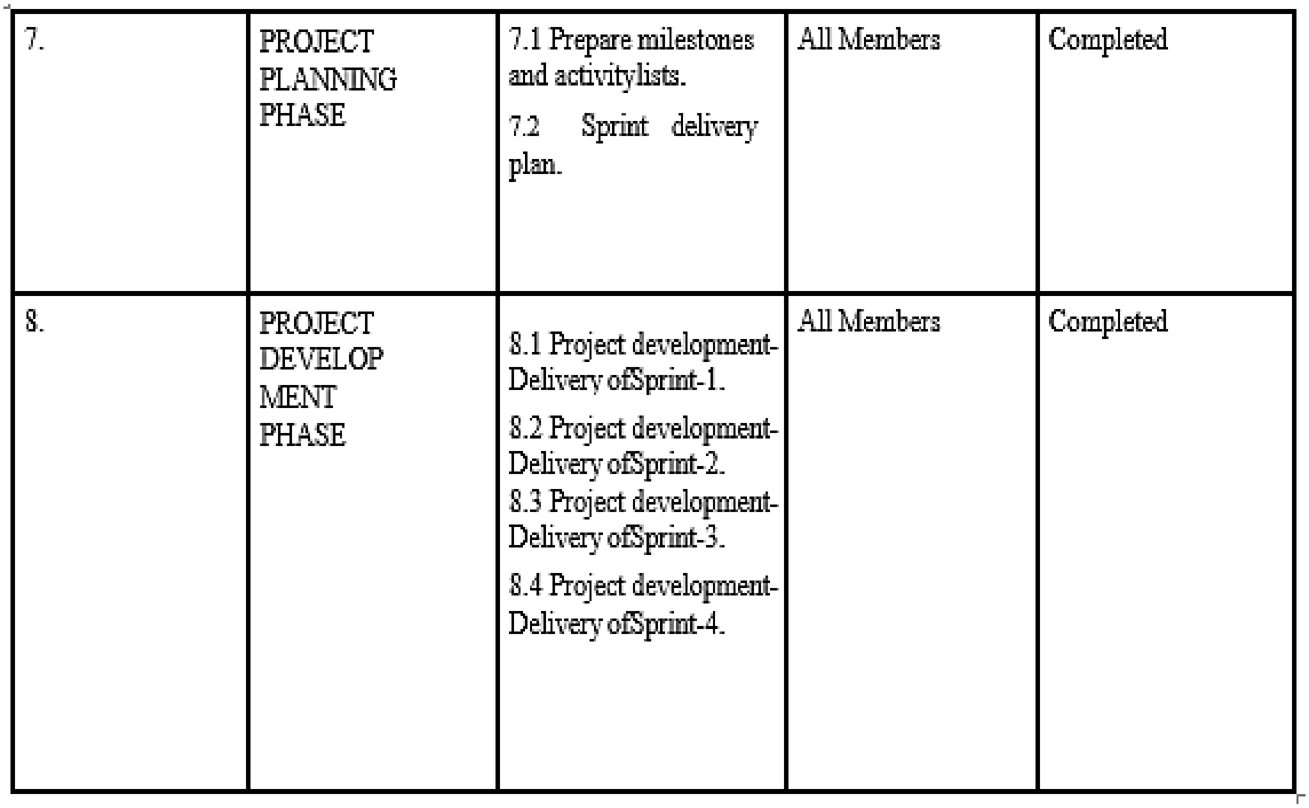


		image to container registry and deploy on Kubernetes		
4.	IDEATION PHASE	4.1 Literature Review. 4.2 Empathy map. 4.3 Ideation.	All Members	Completed
5.	PROJECT DESIGN PHASE – I	5.1 Proposed Solution 5.2 Problem Solution Fit. 5.3 Solution Architecture	All Members	Completed
6.	PROJECT DESIGN PHASE -II	6.1 Customer journey. 6.2 Functional requirement. 6.3 Data flow Diagrams. 6.4 Technology Architecture.	All Members	Completed



7.	PROJECT PLANNING PHASE	7.1 Prepare milestones and activitylists. 7.2 Sprint delivery plan.	All Members	Completed
8.	PROJECT DEVELOP MENT PHASE	8.1 Project development-Delivery ofSprint-1. 8.2 Project development-Delivery ofSprint-2. 8.3 Project development-Delivery ofSprint-3. 8.4 Project development-Delivery ofSprint-4.	All Members	Completed

6.2 Sprint Delivery Plan

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**SprintEnd Date(Planned)**	**Story Points Completed (as on PlannedEnd Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov2022	20	05 Nov2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov2022	20	19 Nov2022

Sprint	Functional Requirement (Epic)	User StoryNumber	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I canregister for theapplication by entering my email, password, and confirming my password.	5	High	
Sprint-1		USN-2	As a user, I will receiveconfirmation email once I haveregistered for the application	5	High	
Sprint-1		USN-3	As a user, I can register for the applicationthrough Gmail	5	Medium	
Sprint-1	Login	USN-4	As a user, I can log into the application byentering email &password	5	High	
Sprint-2	Dashboard	USN-5	As a user, I can enter the interests and choices of newsI want to see for the firsttime indashboard.	10	High	
Sprint-2	Dashboard User Interface	USN - 11	Administrator designing the user interface	10	Medium	
Sprint-3		USN-6	As a userI can go through the feed of newsfiltered according to my wish.	10	High	
Sprint-3		USN-7	Asa user, I can log out my account in settings.	10	Medium	
Sprint-4		USN-8	Asa user, I canupdate my interests and choice in account settings.	10	Medium	
Sprint-4		USN-9	Solve issues brought up by client	5	Medium	
Sprint-4		USN-10	Roll out updates and bug fixes	5	High	

7. CODING & SOLUTIONING

7.1 Feature 1

Create Your Own Account

Login.html

```
!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- ===== Iconscout CSS ===== -->
    <link rel="stylesheet"
href="https://unicons.iconscout.com/release/v4.0.0/css/line.css">

    <!-- ===== CSS ===== -->
    <link rel="stylesheet" href="{{url_for('static',filename='css/style1.css') }}">
    <title>Login & Registration Form</title>
</head>
<script>
    function login1() {
        email= document.getElementById("email").value
        password= document.getElementById("password").value
            fetch('/login?email=' + email + '&password=' + password)
                .then((response) => { return response.json()})
                .then((data) => {
                    console.log(data)
                    if (data.code == 200) {
                        location.href = "/home"
                    }
                    else {
                        alert("Incorrect EMail or Password")
                        location.href = "/"
                    }
                })
                .catch((e) => console.log(e))
        }
```

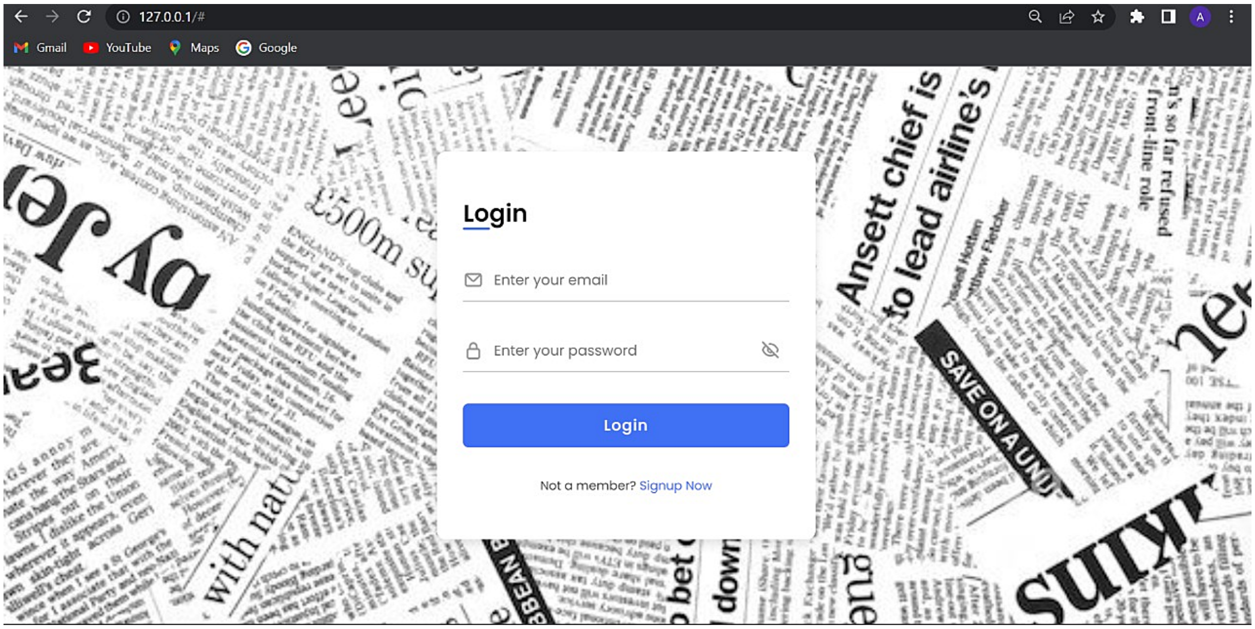
```
function signup() {
    name = document.getElementById('signup_name').value
    email = document.getElementById('signup_email').value
    pass1 = document.getElementById('password1').value
    pass2 = document.getElementById('password2').value

    if (pass1 != pass2) {
      alert(" password doest match")
      return
    }
    else {
      fetch('/signup?name=' + name + '&email=' + email + '&password=' + pass1)
        .then((response) => { return response.json() })
        .then((data) => {
          if (data.code == 200) {
            location.href = '/'
          }
          else {
            alert("please try after sometime")
          }
        })
        .catch((e) => console.log(e))
```

```
            }


        }
</script>
<body>

    <div class="container">
        <div class="forms">
            <div class="form login">
                <span class="title">Login</span>


                    <div class="input-field">
                        <input id="email" type="text" placeholder="Enter your email" required>
                        <i class="uil uil-envelope icon"></i>
                    </div>
                    <div class="input-field">
                        <input id="password" type="password" class="password"
placeholder="Enter your password" required>
                        <i class="uil uil-lock icon"></i>
                        <i class="uil uil-eye-slash showHidePw"></i>
                    </div>
                    <div class="checkbox-text">
                        <div class="checkbox-content">
                            <input type="checkbox" id="logCheck">
                            <label for="logCheck" class="text">Remember me</label>
                        </div>

                        <a href="#" class="text">Forgot password?</a>
                    </div>

                    <div class="input-field button">
                        <input type="button" value="Login"
class="file_submit"onclick=login1()>
                    </div>


                <div class="login-signup">
                    <span class="text">Not a member?
                        <a href="#" class="text signup-link">Signup Now</a>
                    </span>
                </div>
            </div>
```

```
    <!-- Registration Form -->
    <div class="form signup">
        <span class="title">Sign up</span>


            <div class="input-field">
                <input id="signup_name" type="text" placeholder="Enter your name" required>
                <i class="uil uil-user"></i>
            </div>
            <div class="input-field">
                <input id="signup_email" type="text" placeholder="Enter your email" required>
                <i class="uil uil-envelope icon"></i>
            </div>
            <div class="input-field">
                <input id="password1" type="password" class="password" placeholder="Create a password" required>
                <i class="uil uil-lock icon"></i>
            </div>
            <div class="input-field">
```

```
                        <input id="password2" type="password" class="password"
placeholder="Confirm a password" required>
                        <i class="uil uil-lock icon"></i>
                        <i class="uil uil-eye-slash showHidePw"></i>
                    </div>

                    <div class="checkbox-text">
                        <div class="checkbox-content">
                            <input type="checkbox" id="termCon">
                            <label for="termCon" class="text">I accepted all terms and
conditions</label>
                        </div>
                    </div>

                    <div class="input-field button">
                        <input type="button" value="Signup" onclick=signup()>
                    </div>


                <div class="login-signup">
                    <span class="text">Already a member?
                        <a href="#" class="text login-link">Login Now</a>
                    </span>
                </div>
            </div>
        </div>
    </div>


    <script>
        const container = document.querySelector(".container"),
    pwShowHide = document.querySelectorAll(".showHidePw"),
    pwFields = document.querySelectorAll(".password"),
    signUp = document.querySelector(".signup-link"),
    login = document.querySelector(".login-link");

//   js code to show/hide password and change icon
pwShowHide.forEach(eyeIcon => {
    eyeIcon.addEventListener("click", () => {
        pwFields.forEach(pwField => {
            if (pwField.type === "password") {
                pwField.type = "text";

                pwShowHide.forEach(icon => {
                    icon.classList.replace("uil-eye-slash", "uil-eye");
                })
            } else {
                pwField.type = "password";

                pwShowHide.forEach(icon => {
                    icon.classList.replace("uil-eye", "uil-eye-slash");
                })
            }
        })
    })
})
// js code to appear signup and login form
signUp.addEventListener("click", () => {
    container.classList.add("active");
});
login.addEventListener("click", () => {
    container.classList.remove("active");
});
    </script>
</body>
</html>
```

OUTPUT:



Home.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>News application</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```

```
<link rel="stylesheet" href="../static/style1.css">
</head>
<body>
            <div class="container">

                        <div class="input-group mb-3">

    <div class="container-fluid">
      <div class="row">
          <div class="app-container">
              <div class="app-header">
                  <div class="app-header-left">
                      <span class="app-icon"></span>
                      <p class="app-name">News Tracker</p>
                      <form action="/home" method="post">
                                                              <div class="search-wrapper">
                        <input type="text" class="search-input" name="keyword" placeholder="Enter keyword ...." aria-label="Recipient's username" aria-describedby="button-addon2" >
                        <!-- <button class="btn btn-outline-secondary" type="button" id="button-addon2">Search</button> -->
                          <!-- <input class="search-input" type="text" placeholder="Search"> -->
                                                                        </form>
                                                                          <button class="mode-switch" title="Search"><svg
```

```
xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="none"
stroke="currentColor"

                            stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
class="feather feather-search"

                            viewBox="0 0 24 24">



                            <circle cx="11" cy="11" r="8"></circle>

                            <path d="M21 21l-4.35-4.35"></path>

                        </svg>

                      </button>



                      </div>

                  </div>

                  <div class="app-header-right">

                      <button class="mode-switch" title="Switch Theme">

                          <svg class="moon" fill="none" stroke="currentColor" stroke-linecap="round"
stroke-linejoin="round"

                            stroke-width="2" width="24" height="24" viewBox="0 0 24 24">

                            <defs></defs>

                            <path d="M21 12.79A9 9 0 1111.21 3 7 7 0 0021 12.79z"></path>

                          </svg>

                      </button>

                      <button class="loginbtn" title="Logout"
```

```
onclick="location.href='/'">Logout</button>

           </div>

           </div>

     </div>

            </div>


      </div>

      {% if all_headlines %}

      <h2 style="padding-left:45% ;">Headlines</h2>

      <div class="row row-cols-1 row-cols-md-2 g-4 mx-3 my-3">

            {% for headline in all_headlines %}

            <div class="col">

                  <div class="card h-100">

                  <img src="{{headline['urlToImage']}}" class="card-img-top" alt="...">

                  <div class="card-body">

                        <h3 class="card-title">{{headline['title']}}</h3>

                        <p class="card-text">{{headline['description']}} <a
href="{{headline['url']}}" target="blank">Read More...</a></p>



                  </div>

                  <div class="card-footer">
```

```
                <small class="text-muted">{{headline['source']['name']}}</small>
            </div>
            </div>
        </div>
        {% endfor %}
    </div>
    {% endif %}
    {% if all_articles %}
    <h3 style="padding-left:28px ;">Results for '{{keyword}}'</h3>
    <div class="row row-cols-1 row-cols-md-2 g-4 mx-3 my-3">
        {% for article in all_articles %}
        <div class="col">
            <div class="card h-100">
            <img src="{{article['urlToImage']}}" class="card-img-top" alt="...">
            <div class="card-body">
                <h3 class="card-title">{{article['title']}}</h3>
                <p class="card-text">{{article['description']}} <a href="{{article['url']}}" target="blank">Read More...</a></p>



            </div>
            <div class="card-footer">
```

```
                <small class="text-muted">{{article['source']['name']}}</small>
            </div>
            </div>
        </div>
        {% endfor %}
    </div>
    {% endif %}
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous">
</script>
</body>
</html>
```

7.2 Feature 2

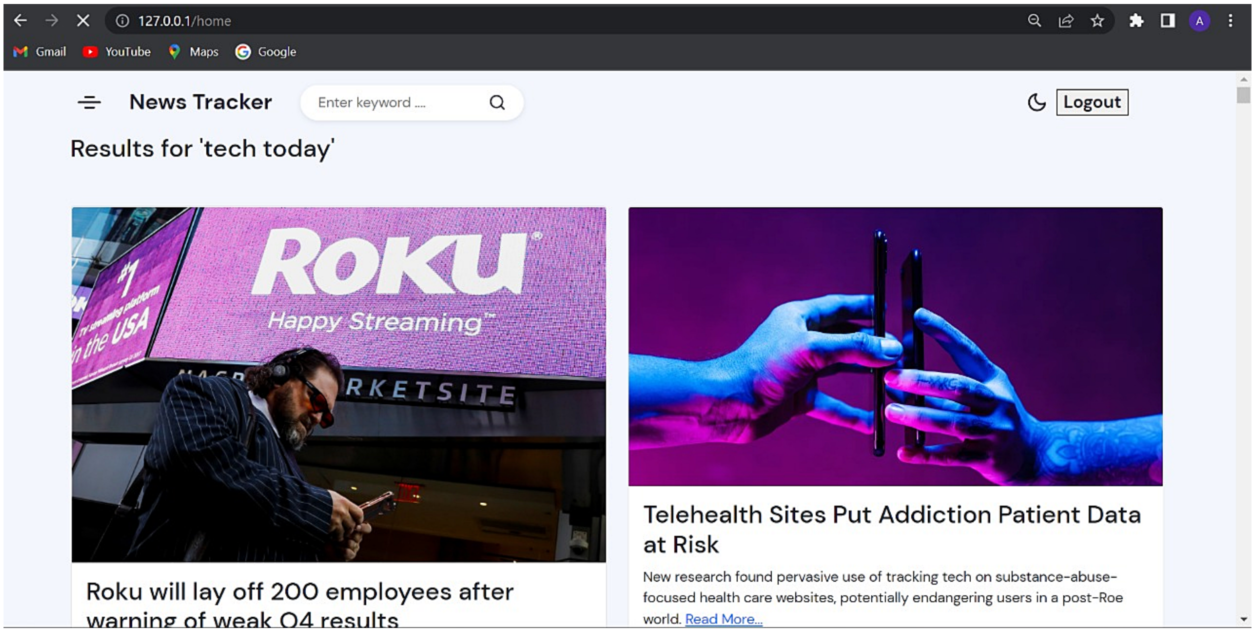
Get Article Related To Search

App.py

```
from flask import Flask, jsonify, render_template, request,url_for,redirect

from newsapi import NewsApiClient

import db

from db import ibm_db



# init flask app

app = Flask(_name_)



# Init news api

newsapi = NewsApiClient(api_key='dbd7a6dece8f43faa88d9a976c066a13')

# helper function

@app.route('/',methods=['GET'])

def loginPage():

    return render_template("login.html")

@app.route("/login" ,methods = ['GET'])

def login():
    global userid
    msg = ''
    if request.method == 'GET':
        email = request.args.get("email")
        password = request.args.get("password")

        sql = "SELECT * FROM USERDETAILS WHERE email = ? AND password = ?"
        stmt = ibm_db.prepare(db.conn,sql)

        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)

        if account:
            return jsonify({"code":200})

        else:
            return jsonify({"code":307})
```

```
@app.route('/signup',methods=['GET'])
def signup():
    email = request.args.get("email")
    password = request.args.get("password")
    name=request.args.get("name")
    name = request.args.get("name")
    email = request.args.get("email")
    password = request.args.get("password")
    sql = "INSERT INTO USERDETAILS (NAME,EMAIL,PASSWORD)
VALUES('"+name+"','"+email+"','"+password+"')"
    stmt = ibm_db.prepare(db.conn,sql)
    ibm_db.execute(stmt)
    return jsonify({"code":200})


def get_sources_and_domains():
    all_sources = newsapi.get_sources()['sources']
    sources = []
    domains = []
    for e in all_sources:
        id = e['id']
        domain = e['url'].replace("http://", "")
        domain = domain.replace("https://", "")
        domain = domain.replace("www.", "")
        slash = domain.find('/')
        if slash != -1:
            domain = domain[:slash]
        sources.append(id)
        domains.append(domain)
    sources = ", ".join(sources)
    domains = ", ".join(domains)
    return sources, domains

@app.route("/home", methods=['GET', 'POST'])
def home():
    if request.method == "POST":
        sources, domains = get_sources_and_domains()
        keyword = request.form["keyword"]
        related_news = newsapi.get_everything(q=keyword,
                                              sources=sources,
                                              domains=domains,
                                              language='en',
                                              sort_by='relevancy')
        no_of_articles = related_news['totalResults']
        if no_of_articles > 100:
            no_of_articles = 100
        all_articles = newsapi.get_everything(q=keyword,
                                    sources=sources,
                                    domains=domains,
                                    language='en',
                                    sort_by='relevancy',
                                    page_size =
no_of_articles)['articles']
        return render_template("home.html", all_articles = all_articles,
                            keyword=keyword)
    else:
        top_headlines = newsapi.get_top_headlines(country="in", language="en")
        total_results = top_headlines['totalResults']
        if total_results > 100:
            total_results = 100
        all_headlines = newsapi.get_top_headlines(country="in",

language="en",

page_size=total_results)['articles']
        return render_template("home.html", all_headlines = all_headlines)
    return render_template("home.html")


if _name_ == "_main_":
    app.run(debug=True,host='0.0.0.0', port=80)
```

OUTPUT:



News Tracker

Results for 'tech today'

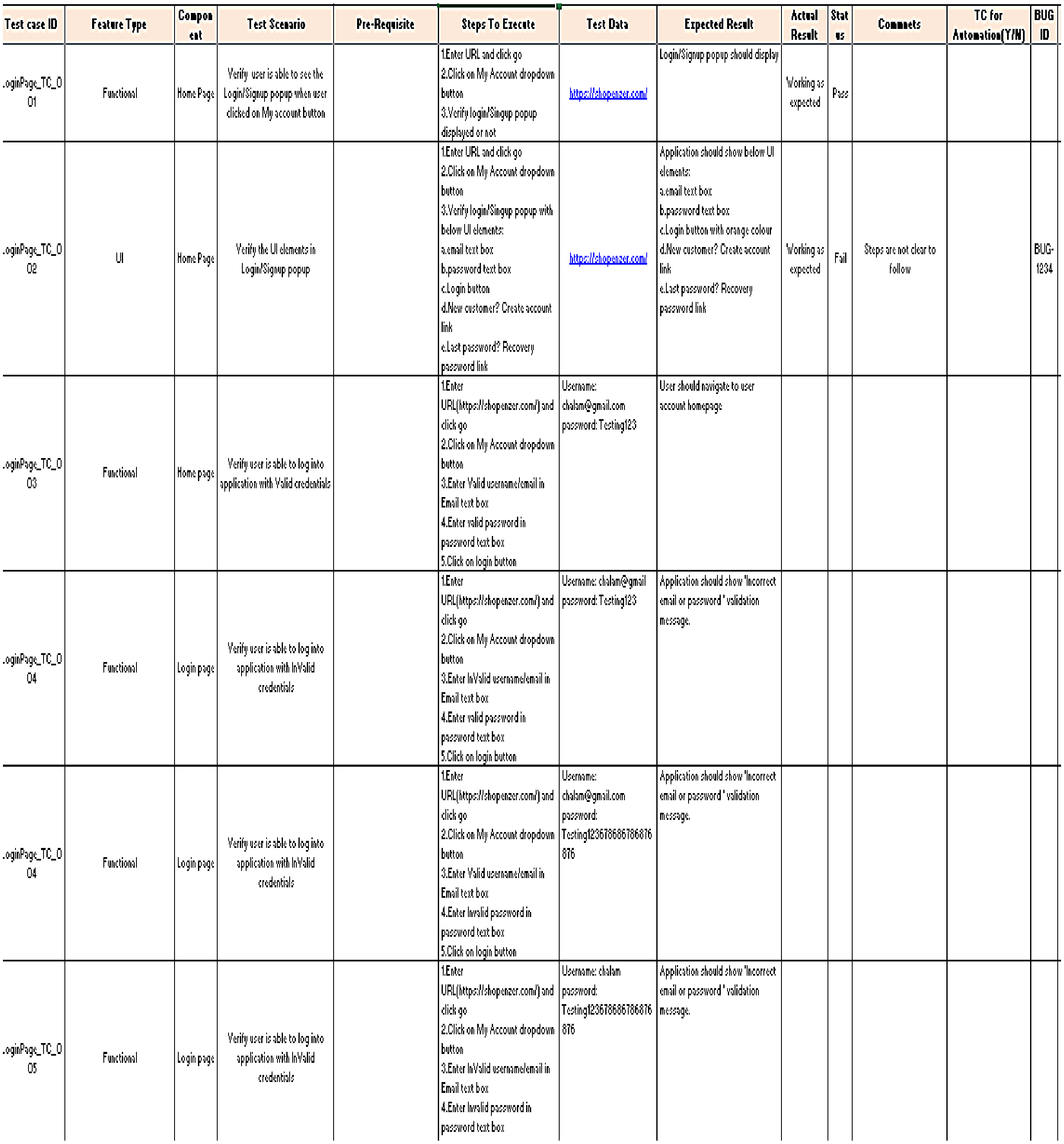
Roku will lay off 200 employees after warning of weak Q4 results

Telehealth Sites Put Addiction Patient Data at Risk

New research found pervasive use of tracking tech on substance-abuse-focused health care websites, potentially endangering users in a post-Roe world. Read More...

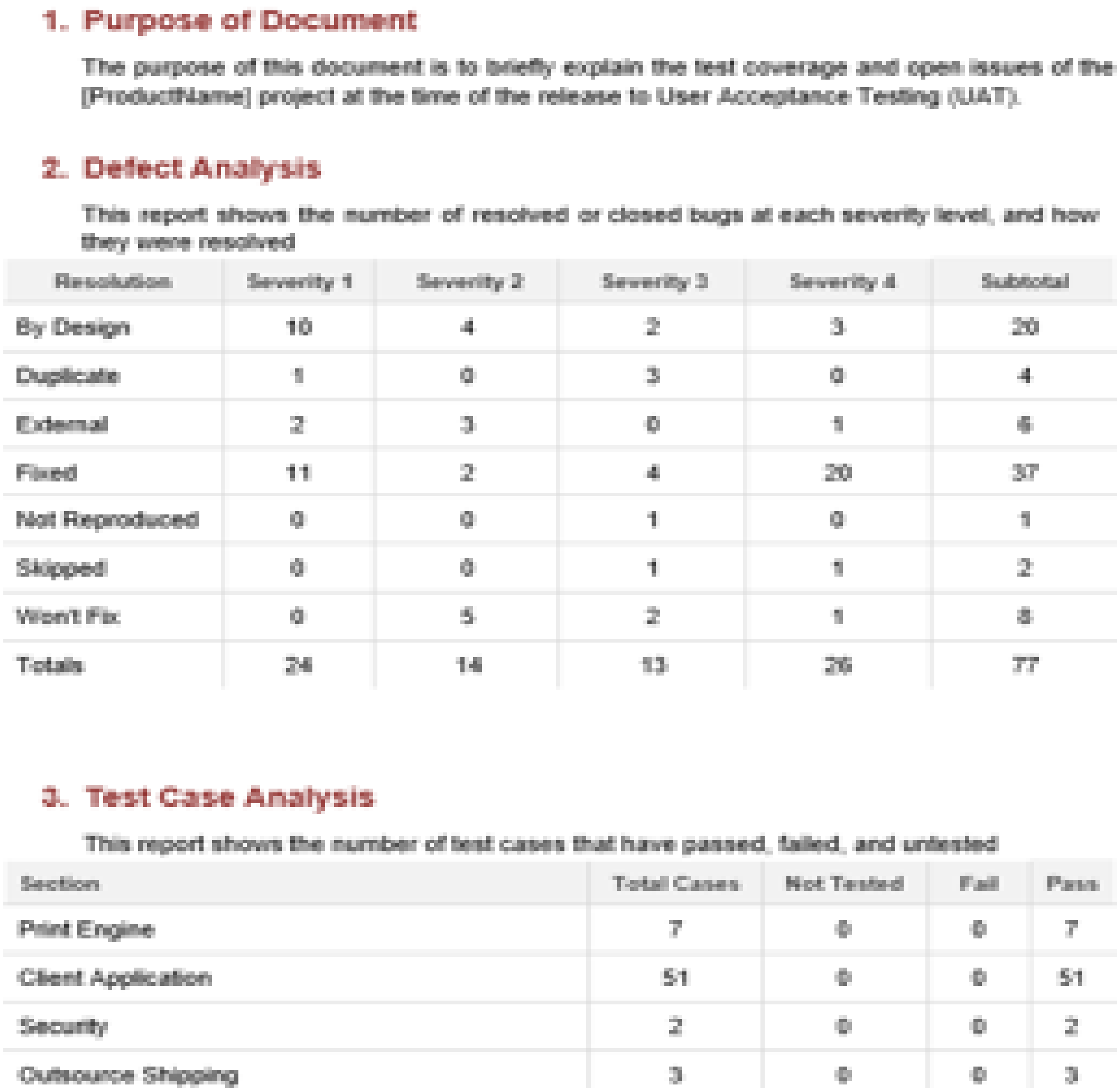
8. Testing

8.1 Test Cases



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID
.oginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://shopenzer.com/	Login/Signup popup should display	Working as expected	Pass			
.oginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://shopenzer.com/	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Fail	Steps are not clear to follow		BUG-1234
.oginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage					
.oginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.					
.oginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.					
.oginPage_TC_005	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box	Username: chalam password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.					

8.2 Acceptance Testing



1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

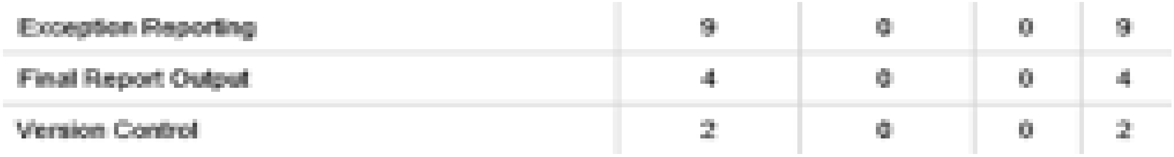
2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
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	77

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. Results

9.1 Performance Metrics

Display the most relevant story first

It is an important step to display the most recent or top story first to your app users.

If you have one article as the primary focus instead of having a carousel with multiple options, it will help you increase your Click-though rate (CTR) to that article.

Carousels are often ignored while functional, and your app users will only get to see the first frame in most cases.

Notre Dame University tested a carousel and found that only 1% of their total visitors clicked on it. 84% of those visitors only interacted with the very first slide.

Rather than introducing a carousel, draw your user's attention to the most famous story of the day, when they visit your news app.

Track your changes to page views for your top post, which will increase when more and more people are clicking through to your top story on the app.

Help users find content with categories

If your app readers want to access different types of content, you need to provide an easy way to them. Adding 'content categories' to your main app menu is the perfect way to do this.

A ‘categories section’ will not take up much space on your app screen. However, it offers users a simple and effective way to navigate around your news app.

Once you add the categories section to the app menu, you will find an increase in page views. Instead of sticking to one content category, your app readers will end up discovering the content in different categories that interest them.

Make ads user-friendly

The advertising methods, such as native in-feed advertisements are UX-friendly that helps you generate more revenue, and provide a better user experience to your app readers.

In-feed advertisements are user-friendly as they match and fit seamlessly with your existing content style.

Well, the differences are minor. However, you need to be transparent with your users. For instance, if you include sponsored content, make sure to label them as Sponsored.

Where the regular posts include an author name, the in-feed advertisement shows ‘Sponsored.’

As in-feed ads are user-friendly, they are effective ways to generate revenue. Similarly, they maintain the unique styling of your mobile app and does not hurt the app user experience.

With user-friendly ads, you can see the Click-Through Rate on your ads will increase. You can even expect your users to spend more time on the page if the ads are of high-quality and are relevant.

Leverage the estimated reading time to increase time spent on page

The average adult spends almost 5-6 hours with digital media per day, and only part of this time will be spent reading your content. Incorporating an ‘estimated reading time indicator’ to your news app UX can have a substantial effect.

You need to be transparent about the length of your article. With this transparency, your app audience is likely to commit to reading them. Even if your articles are long, your readers will not get away without reading them.

A study shows that content publishers who regularly post their content over 1000+ words, their bounce rate has dropped by 13%. It means a simple change can lead to a significant traffic increase to your news app.

Find a host of WordPress plugins that enable you to add this feature to your app easily. With this feature, you can expect your user base to increase as now they are more likely to commit to reading the articles.

10. Advantage

1. Information Goes Paperless

Every piece of update you receive is through electronic media hence saving the environment. A tree flourishes for every paper saved. With apps available for every newspaper, Media companies can contribute to utilizing natural resources sustainably.

2. Updates on the Go

Digital News Media allows you to select the type of news you are into, and these updates are just a tap away. News Apps provide you relevant content as per your preference that you can access with ease anywhere and at any time of the day. So, don’t worry if your newspaper guy does not show up at times.

3. Revenue Model:

A News App is a brilliant revenue model. Daffodil has worked with clients who have been apprehensive of the initial cost of App development. But when it comes to a News App, one can easily avoid the unnecessary frills and reduce App cost to a minimum and adding the revenue through in-App purchases and Google Adsense, a news company can make significant profit.

If a Company has done enough ground work such as Brand logo, images & content then App developers will quote a significantly low cost for a News App development and also be done with the development at a quick pace.

4. Interest Generator:

With a News App, a News & Media company will be omnipresent with their readers. 70% of young adults receive their news through a smart phone; A News App will be an easy way to showcase your content, media, news and stories to them.

Whenever the reader pleases, he/she can use your News App as a one-stop point to get all the information they need. And whenever content is updated on the News App, the reader will be rolled towards your App with update notifications.

5. Promotion Tool:

Consider a sensational News cover, with either eloquent background or high national importance, now consider the news going cold because the reader never got notified or a competitor did better promoting. A News App can be a ground-wrecking tool to promote such News activities.

Promoting News subscription services with exclusive e-papers, audio & video clips etc. can be done using the News App as well. All big & hefty companies are trending and tending to direct audiences towards mobile by offering cheaper rates and offers on Apps.

This just goes to show how impactful an App is today.

6. Engagement portal:

News companies can engage with their reader through News App, here comes the harvesting of the reader's psyche by using our nearest & dearest friend, social media. News Apps are usually linked with social media accounts such as Facebook, Twitter, Google+ etc.

Account linking through social media gives access to identifying the reader's behavior on the internet. User can be engaged by being showcasing presence on news feeds. No one can avoid reading a great piece of News especially when the News is always at the tip of their fingertips (literally).

I spoke of the 'Mobilegeddon' before. I must mention that one can totally dodge the tremor of this conjectural storm with the use of News App. Redesigned Google search algorithms will cause major mobile sites to not show up on Google Search, one can use this as an advantage to lure in audience at the expense of competitors falling prey to the 'Mobilegeddon'.

The 'RIPE' theory is but a cup of water from an ocean of opportunities a News App can provide at the behest of the company.

Above being said, I must also mention and state with pronounced eminence that an App can perform well only if the Company & the Developer both work in tandem. It is like a bicycle with two pedals and we at Daffodil believe in keeping both the pedals running for you.

Disadvantage

1. Health Alert:

The excessive use of the internet may cause diseases. Man can get so addicted to the internet that he becomes inseparable from it. The radiation emitted through mobile and laptop are very dangerous for humans as it may cause brain cancer and maybe other illnesses as well.

2. Technical Issues:

The site could crash if a lot of people try accessing it at the same time. If the site crashes, it doesn’t open for a short while then. Also, other technical issues can occur on the website. Those can result in going offline temporarily. Ultimately, this will lead to no access to the site and the offline competitors may gain benefit.

3.Paid Subscription for Daily News:

There are some news sites available online who allow access to readers only after they pay for daily news. Revenue is generated by the site if people subscribe for daily news. Readers may disagree to pay extra money as there are other news websites which are available to read recent news.

11. Conclusion

News will be displayed in short format with title,
image and little description in list view. It will help user to access required news faster.
Search Option: User will be able to search from not only one source but many different sources available within API.

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. If API can’t reach to certain article source it gives null value which can cause problem in JSON parsing.

13. Appendix

Source code link: https://drive.google.com/drive/folders/1tl5E4BjIN0LUosNbqSow6nT3_b6HmUck?usp=sharing

Git hub link: https://github.com/IBM-EPBL/IBM-Project-1233-1658379958

Demo link: https://drive.google.com/file/d/1-3hKtNGg3FKG6_GzK4jG-Jw67k-Huj3p/view?usp=sharing