

# **NEWS TRACKER APPLICATION**

**Domain : Cloud App Development**

**Project Report**

**Submitted by,**

**Team ID : PNT2022TMID19738**

**Team Leader : SabariRohith N**

**Team Member 1 : Kavinayaa N**

**Team Member 2 : Sreeram KT**

**Team Member 3 : Sabareesan A**

**Team Member 4 : Dharanya G**

# **TABLE OF CONTENTS**

- 1. INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
- 2. LITERATURE SURVEY**
  - 2.1 Existing problem
  - 2.2 References
  - 2.3 Problem Statement Definition
- 3. IDEATION & PROPOSED SOLUTION**
  - 3.1 Empathy Map Canvas
  - 3.2 Ideation & Brainstorming
  - 3.3 Proposed Solution
  - 3.4 Problem Solution fit
- 4. REQUIREMENT ANALYSIS**
  - 4.1 Functional requirement
  - 4.2 Non-Functional requirements
- 5. PROJECT DESIGN**
  - 5.1 Data Flow Diagrams
  - 5.2 Solution & Technical Architecture
  - 5.3 User Stories
- 6. PROJECT PLANNING & SCHEDULING**
  - 6.1 Product backlog, Sprint Schedule and Estimation
  - 6.2 Project tracker, Velocity and Burndown chart
- 7. CODING & SOLUTIONING (Explain the features added in the project along with code)**
  - 7.1 Server Side
  - 7.2 Client Side
  - 7.3 Schema
- 8. TESTING**
  - 8.1 Acceptance Testing
  - 8.2 Test Cases Report
- 9. RESULTS**
  - 9.1 Sample Screens
- 10. ADVANTAGES & DISADVANTAGES**
  - 10.1 Advantages
  - 10.2 Disadvantages
- 11. FUTURE SCOPE**
- 12. REFERENCES**

# **INTRODUCTION**

## **1.1 Project Overview**

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, etc.... as on official websites.

## **1.2 Purpose**

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. 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. A user can read news only from their interested fields rather than reading all the news. This application provides users with a trusted and secured ecosystem. News shared through the application is original and spam free. This application encourages its users to provide feedback. Based on that feedback, developments were made eventually.

## **LITERATURE SURVEY**

### **2.1 Existing Solutions**

(a) Opinion Extraction, Summarization and Tracking inNews - 2003

Authors : Lun Wei Ku, Yu Ting Liang, Hsin hsi Chen

In this paper, both news and web blog articles are investigated. TREC, NTCIR and articles collected from web blogs serve as the information sources for opinion extraction. 85% Documents related to the issue of animal cloning are selected as the experimental materials. Algorithms for opinion extraction at word, sentence and document level are proposed. The issue of relevant sentence selection is discussed, and then topical and opinionated information are summarized. Opinion summarizations are visualized by representative sentences

(b) Breaking News Detection and Tracking in Twitter - 2010

Authors : Swit Phuvioada wat, Tsuyoshi Murata

Twitter has been used as one of the communication channels for spreading breaking news. We propose a method to collect, group, rank and track breaking news in Twitter. Since short length messages make similarity comparison difficult, we boost scores on proper nouns to improve the grouping results. Each group is ranked based on popularity and reliability factors.

(c) News Classification based on their Headlines - 2014

Authors : Mazhar Iqbal Rana

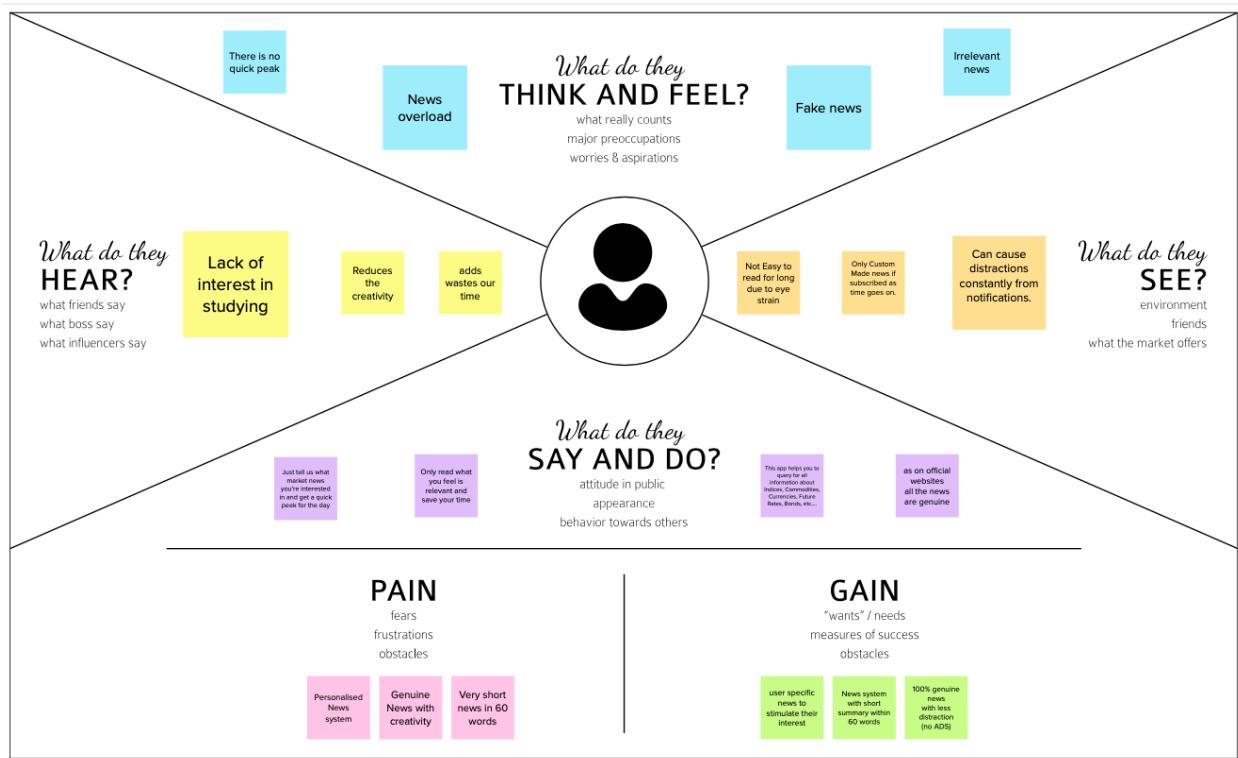
Focus on full text classification i.e. full news, huge documents, long length texts etc. is more prominent as compared to the short length text. In this paper Existing classifiers and their working methodologies are being compared and 76% results are presented effectively

## **2.2 Proposed Solution**

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. 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. A user can read news only from their interested fields rather than reading all the news. This application provides users with a trusted and secured ecosystem. News shared through the application is original and spam free. This application encourages its users to provide feedback. Based on that feedback, developments were made eventually. Add advertisements to the application, so that we can get revenue from those advertisement-sponsored organizations. More advertisements may irritate the user. Add premium subscription, users who subscribe for premium won't get advertisements. As it was an application-based project, correct ideation and execution can develop an application with no bugs and errors, so that the user might like our application and some might suggest and share it to their surroundings, resulting in an increase in our application insights.

# IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

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!

#### Kavinayaa N

Giving an option to choose he font size for people prefers big fonts	Apps generate income through subscription	App size should be small
Providing the details of the source from which the news has been collected so as a reliable source to compare it with.		Create a Like and Dislike feature in the app.

#### SabariRohith N

Audio features to discuss about news	Show the percentage of the News truthfulness
Summarize the news by day, week, month, Year.	Users don't want to spend time reading the entire content. They need short and crisp news

#### Sabareesan A

Providing credible links for the user to refer to in case of any doubts on news	Add remainder for General events in Society	Provide the TOP STORIES section to gain users' attention.
Making sure that the news will be in every language and covers the national and international news well.		Provide the news from all the trusted worldwide sources, and then mention which source has selected the news. By assuming that users can't differentiate between fake news.

#### Dharanya G

Adding the duration of the article to help users decide if they want to spend time reading it	User can earn points for spending more or reading more article in this app
Ask the user to set up Do Not Disturb Time. During this time, do not push any notifications to the user	Upvotes, Downvotes and Reactions

#### Sreeram KT

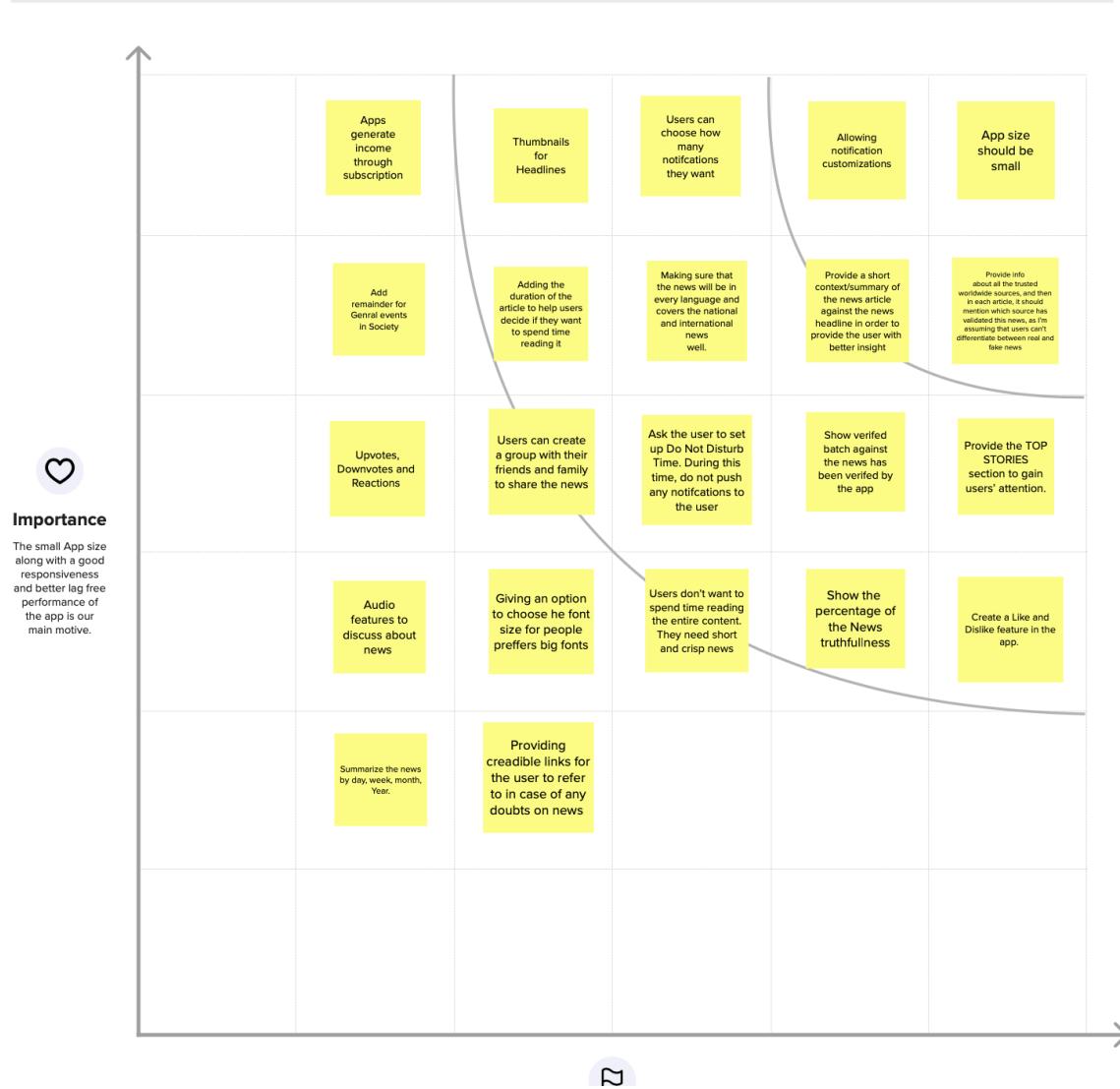
Show verified batch against the news has been verified by the app	Users can choose how many notifications they want	Provide a short content summary of the news article against the news headline in order to provide the user with better insight
Allowing notification customizations		Add remainder for General events in Society

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



### **3.3 Proposed Solution**

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. 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. A user can read news only from their interested fields rather than reading all the news.

This application provides users with a trusted and secured ecosystem. News shared through the application is original and spam free. This application encourages its users to provide feedback. Based on that feedback, developments were made eventually. Add advertisements to the application, so that we can get revenue from those advertisement-sponsored organizations. More advertisements may irritate the user. Add premium subscription, users who subscribe for premium won't get advertisements. As it was an application-based project, correct ideation and execution can develop an application with no bugs and errors, so that the user might like our application and some might suggest and share it to their surroundings, resulting in an increase in our application insights.

## 3.4 Problem Solution Fit

**Problem-Solution fit canvas 2.0**

Purpose / Vision			
<b>1. CUSTOMER SEGMENT(S)</b> Who is your customer? i.e. working parents of 0-5 y.o. kids  1.News reader 2.People	<b>6. CUSTOMER CONSTRAINTS</b> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.  1.It will consume more time 2.It will consume more cost 3.Network connection	<b>5. AVAILABLE SOLUTIONS</b> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking  People may use either newspaper or social media or youtube channels to know the news	<b>AS</b> Explore AS, differentiate
<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.  1.People can get simultaneous breaking news 2.We can avoid fake news 3.News received at correct time	<b>9. PROBLEM ROOT CAUSE</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.  In a busy world people not have allocate time for reading newspaper and watching news channels	<b>7. BEHAVIOUR</b> What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)  People follow youtube channels but this will not possible to know all news. People buy a news paper they don't read all news because of time cons	<b>BE</b> Focus on J&P, tap into BE, understand RC
<b>3. TRIGGERS</b> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.  Reading about a more efficient solution in the news	<b>10. YOUR SOLUTION</b> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits really. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.  Making separate space for each category of news. people select the news category and know all news about that.	<b>8. CHANNELS OF BEHAVIOUR</b> 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 In online people know news faster through network	<b>CH</b> Extract online & offline CH of BE
<b>4. EMOTIONS: BEFORE / AFTER</b> How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.  People will know the news in faster		8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. In offline people must allocate time for reading newspaper	

 Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license  
Created by Daria Nepriakhina / Amaltama.com



# REQUIREMENT ANALYSIS

## 4.1 Functional Requirements

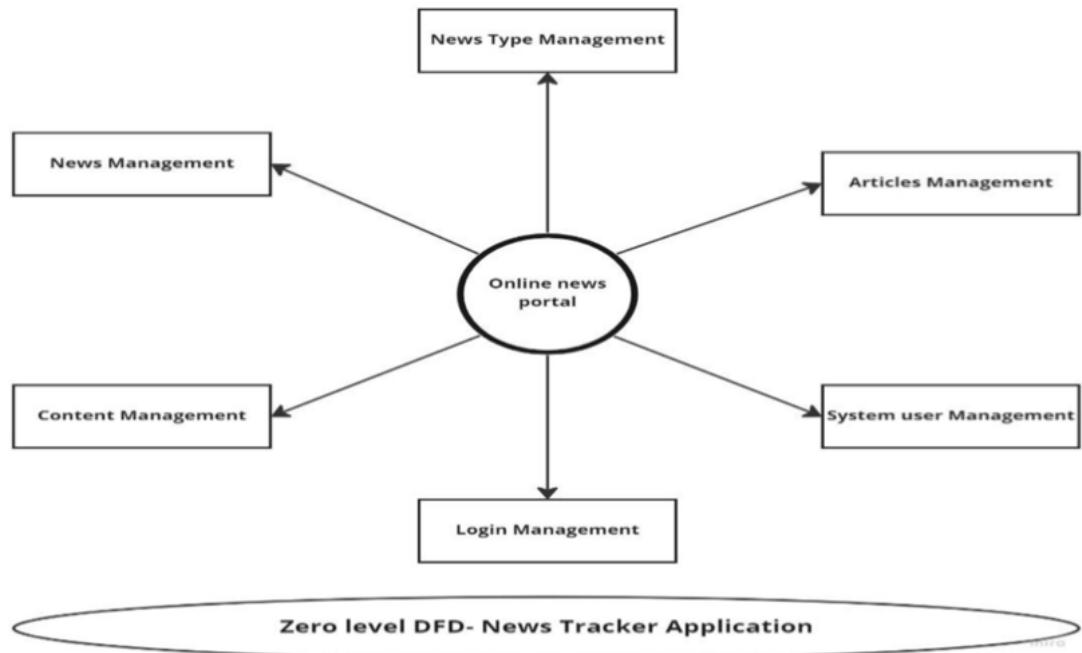
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Installation	Install through playstore or website
FR-4	User Login	Login using username and password

## 4.2 Non-Functional Requirements

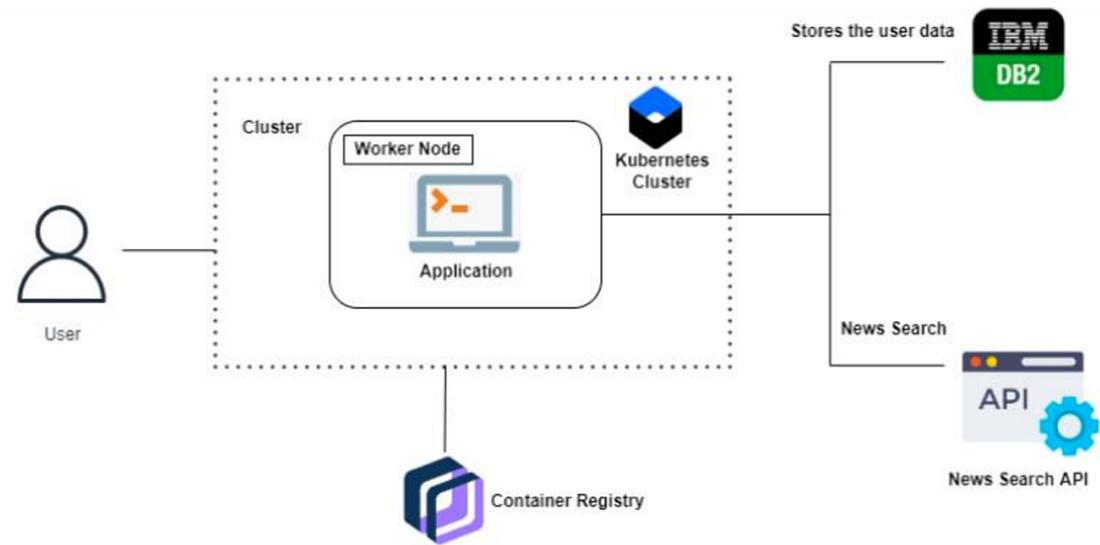
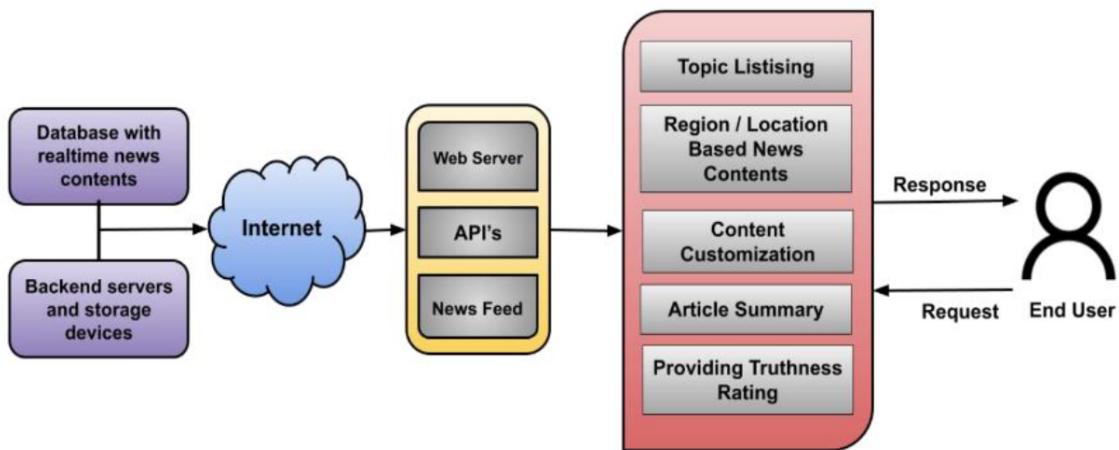
FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	User friendly UI so that anyone can use it easily
NFR-2	<b>Security</b>	Safer App and not allows fake news
NFR-3	<b>Performance</b>	Faster response to user command and is more efficient when comes to performance
NFR-4	<b>Availability</b>	More features are added to the app on every update sprints
NFR-5	<b>Reliability</b>	Due to the safer environment, it is reliable
NFR-6	<b>Scalability</b>	Scalable in nature with efficient performance

# PROJECT DESIGN

## 5.1 Data Flow Diagrams



## 5.2 Solution & Technical Architecture



### 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Registration	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
Customer (Mobile user)	Registration	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
Customer (Mobile user)	Registration	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
Customer (Mobile user)	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access the dashboard with registered mail Id and password	High	Sprint-1
Customer (Mobile user)	Dashboard	USN-6	Feeds on the basis of user interest	I can see news based on my interest	Medium	Sprint 1
Customer (Web user)	Browsing	USN-7	Ability to browse on a news topic	I can search for a news	Medium	Sprint 1
Administrator	Validation	USN-8	Maintain and prevent fake news	I can validate the news	High	Sprint 1
Administrator	Login	USN-9	Login functionality in browser	I can login to my account	High	Sprint 1

# **PROJECT PLANNING AND SCHEDULING**

## **6.1 Product Backlog, Sprint Schedule, and Estimation**

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sreeram KT
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Kavinayaa N
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Medium	Dharanya G
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	SabariRohith N
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Sabareesan A
	Dashboard		As a user, I can search and get suggestions on current news feeds	2	High	Kavinayaa N
Sprint-3	Notification	USN-6	As a user, I can receive notifications on trending and subscribed news articles	2	Medium	Sreeram KT
Sprint-4	Updates	USN-7	As a user, I can get updated and latest news whenever I use this service	2	High	SabariRohith N Sabareesan A

## **6.2 Project Tracker, Velocity & Burndown Chart**

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

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

For Sprint-1 the Average Velocity (AV) is:  $AV = \text{Sprint Duration} / \text{velocity} = 6 / 6 = 1$

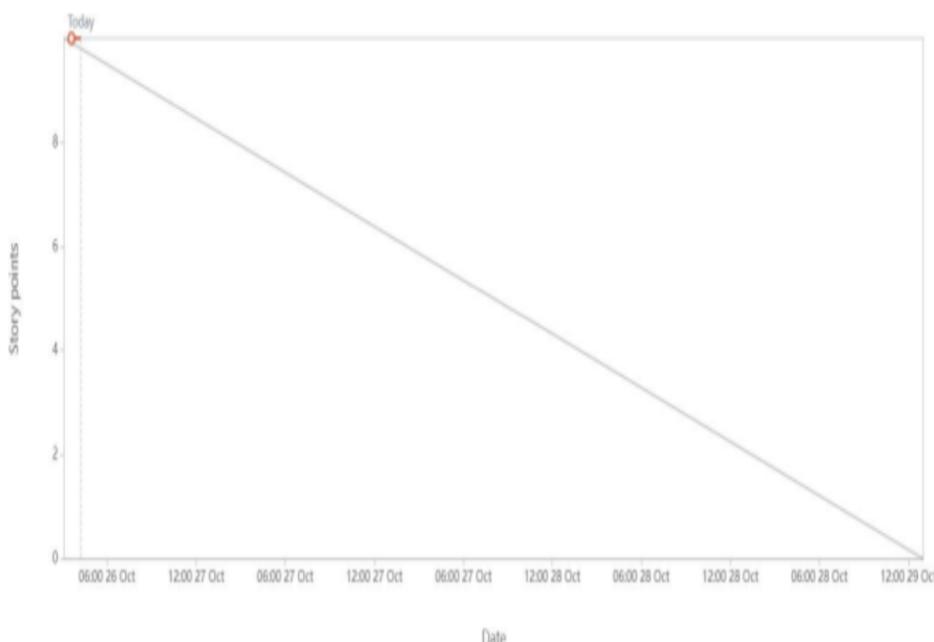
For Sprint-2 the Average Velocity (AV) is:  $AV = \text{Sprint Duration} / \text{velocity} = 18 / 6 = 3.0$

For Sprint-3 the Average Velocity (AV) is:  $AV = \text{Sprint Duration} / \text{velocity} = 16 / 6 = 2.6$

For Sprint-4 the Average Velocity (AV) is:  $AV = \text{Sprint Duration} / \text{velocity} = 12 / 6 = 2.0$

Average Velocity =  $14 / 6$  AV = 2.333

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



# **CODING & SOLUTIONING**

## **7.1 Client Side**

### **about.html**

```
<!doctype html> <html lang="en"> <head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
<!-- Custom Styles -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
<title>NewsToday About</title>
</head>
<body>
<nav class="navbar navbar-expand-md">
<a class="navbar-brand" href="/"><h2 style="font-family:'Source Code Pro', monospace;">NewsToday</h2></a>
<button class="navbar-toggler navbar-light" type="button" data-toggle="collapse" data-target="#main-navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="main-navigation">
<ul class="navbar-nav"> <li class="nav-item">
<a class="nav-link" href="/">Home</a> </li>
<li class="nav-item">
```

```

<a class="nav-link" href="/sources">Sources</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/about">About me</a> </li>&nbsp;&nbsp;
<form action="{{ url_for('search') }}", method="post">
<input type="text" name="searchBar" placeholder="Search news here">
<button>Search</button>
</form>
</ul>
</div>
</nav>
<div class="jumbotron">
<h1 class="display-4">Team Members</h1> <hr class="my-4">
<ol><li>SabariRohith N - Team Leader</li>
<li>Kavinayaa N</li>
<li>Sreeram KT</li>
<li>Sabareesan A</li>
<li>Dharanya</li>
</ol>
</div>
</body>
</html>

```

## index.html

```

<!doctype html> <html lang="en"> <head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"> <!-- Custom
Styles -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link rel="preconnect" href="https://fonts.gstatic.com">

```

```
<link
  href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
  rel="stylesheet">
  <title>NewsToday Homepage</title>
</head>
<nav id="navbar" class="navbar navbar-expand-md">
  <a class="navbar-brand" href="/"><h2 style="font-family:'Source Code Pro', monospace;">NewsToday</h2></a>
  <button class="navbar-toggler navbar-light" type="button" data-toggle="collapse" data-target="#main-navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="main-navigation">
    <ul class="navbar-nav mr-auto"> <img class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="/">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/sources">Sources</a> </li>
      <li class="nav-item">
        <a class="nav-link" href="/about">About me</a>
      </li>&nbsp;&nbsp;
      <form action="{{ url_for('search') }}", method="post">
        <input type="text" name="searchBar" placeholder="Search news here">
        <button>Search</button>
      </form>
    </ul>
  </div>
</nav>
<div class="container"> <h5>Category</h5>
  <ul class="nav nav-tabs" role="tablist">
    <li role="presentation"><a href="#business" role="tab" data-toggle="tab">Business&nbsp;&nbsp;&nbsp;</a></li>
    <li role="presentation"><a href="#entertainment" role="tab" data-toggle="tab">Entertainment&nbsp;&nbsp;&nbsp;</a></li>
  2 <li role="presentation" class="active"><a href="#general" role="tab" data-
```

```

toggle="tab">General &nbsp;&nbsp;&nbsp;</a></li>
<li role="presentation"><a href="#health" role="tab" data-
toggle="tab">Health &nbsp;&nbsp;&nbsp;</a></li>
<li role="presentation"><a href="#science" role="tab" data-
toggle="tab">Science &nbsp;&nbsp;&nbsp;</a></li>
<li role="presentation"><a href="#sports" role="tab" data-
toggle="tab">Sports &nbsp;&nbsp;&nbsp;</a></li>
<li role="presentation"><a href="#technology" role="tab" data-
toggle="tab">Technology &nbsp;&nbsp;&nbsp;</a></li>
</ul>
</li>
<br><br>
<!-- Tab panes -->
<div class="tab-content">
  <div role="tabpanel" class="tab-pane" id="business">
    {% for i in response1['articles'] %} <div class="card">
       <div
        class="card-body">
        <h5 class="card-title">{{ i['title'] }}</h5>
        <p class="card-text"> {{ i['description'] }}</p>
        <a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
      </div>
    </div>
    {% endfor %}
  </div>
  <div role="tabpanel" class="tab-pane" id="entertainment">
    {% for i in response2['articles'] %} <div class="card">
       <div
        class="card-body">
        <h5 class="card-title">{{ i['title'] }}</h5>
        <p class="card-text"> {{ i['description'] }}</p>
        <a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
      </div>
    </div>
    {% endfor %}
  </div>
  <div role="tabpanel" class="tab-pane active" id="general">

```

```
{% for i in response3['articles'] %} <div class="card">
 <div
class="card-body">
<h5 class="card-title">{{ i['title'] }}</h5>
<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
</div>
</div>
{% endfor %}
</div>

3 <div role="tabpanel" class="tab-pane" id="health">
{% for i in response4['articles'] %} <div class="card">
 <div
class="card-body">
<h5 class="card-title">{{ i['title'] }}</h5>
<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
</div>
</div>
{% endfor %}
</div>

<div role="tabpanel" class="tab-pane" id="science">
{% for i in response5['articles'] %} <div class="card">
 <div
class="card-body">
<h5 class="card-title">{{ i['title'] }}</h5>
<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
</div>
</div>
{% endfor %}
</div>

<div role="tabpanel" class="tab-pane" id="sports">
{% for i in response6['articles'] %} <div class="card">
 <div
class="card-body">
<h5 class="card-title">{{ i['title'] }}</h5>
```

```

<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
</div>
</div>
{% endfor %}
</div>
<div role="tabpanel" class="tab-pane" id="technology">
{% for i in response7['articles'] %} <div class="card">
 <div
class="card-body">
4 <h5 class="card-title">{{ i['title'] }}</h5>
<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a> </div>
</div>
{% endfor %}
</div>
</div>
</body>
</html>

```

## search.html

```

<!doctype html> <html lang="en"> <head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"> <!-- Custom
Styles -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
<title>NewsToday sources</title>

```

```
</head>
<body>
<nav class="navbar navbar-expand-md">
<a class="navbar-brand" href="/"><h2 style="font-family:'Source Code Pro', monospace;">NewsToday</h2></a>
<button class="navbar-toggler navbar-light" type="button" data-toggle="collapse" data-target="#main-navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="main-navigation">
<ul class="navbar-nav"> <li class="nav-item">
<a class="nav-link" href="/">Home</a> </li>
<li class="nav-item">
<a class="nav-link" href="/sources">Sources</a>
</li>
<li class="nav-item"> Type your text
Type your text
Type your text
Type your text
6
<a class="nav-link" href="/about">About me</a> </li>&ampnbsp&ampnbsp
<form action="{{ url_for('search') }}", method="post">
<input type="text" name="searchBar" placeholder="Search news here">
<button>Search</button>
</form>
</ul>
</div>
</nav>
{% for i in response['articles'] %}>
<div class="card">
 <div class="card-body">
<h5 class="card-title">{{ i['title'] }}</h5>
<p class="card-text"> {{ i['description'] }}</p>
<a href="{{ i['url'] }}" class="btn btn-primary">Read more...</a>
</div>
```

```
</div>
{%
endfor
}
</body>
</html>
```

## [sources.html](#)

```
<!doctype html> <html lang="en"> <head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}"> <!-- Custom
Styles -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
<title>NewsToday sources</title>
</head>
<body>
<nav class="navbar navbar-expand-md">
<a class="navbar-brand" href="/"><h2 style="font-family:'Source Code Pro',
monospace;">NewsToday</h2></a>
<button class="navbar-toggler navbar-light" type="button" data-toggle="collapse" data-
target="#main-
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="main-navigation">
7
<ul class="navbar-nav"> <li class="nav-item">
<a class="nav-link" href="/">Home</a> </li>
<li class="nav-item">
```

```

<a class="nav-link" href="/sources">Sources</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/about">About me</a> </li>&nbsp;&nbsp;
<form action="{{ url_for('search') }}", method="post">
<input type="text" name="searchBar" placeholder="Search news here">
<button>Search</button>
</form>
</ul>
</div>
</nav>
{% for i in name['sources'] %}
<div class="card">
<div class="card-header">Category : {{ i['category'] }}, Country : {{ i['country'] }} </div>
<div class="card-body">
<h5 class="card-title">{{ i['name'] }} </h5>
<p class="card-text">{{ i['description'] }} </p>
<a href="{{ i['url'] }}" target='_blank' class="btn btn-primary">Visit site</a>
</div>
</div>
{% endfor %}
</body>
</html>

```

## 404.html

```

<!doctype html> <html lang="en"> <head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
<!-- Custom Styles -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-

```

MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"  
crossorigin="anonymous">

```
<link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
<title>NewsToday 404</title>
</head>
<body>
<nav class="navbar navbar-expand-md">
<a class="navbar-brand" href="/"><h2 style="font-family:'Source Code Pro', monospace;">NewsToday</h2></a>
<button class="navbar-toggler navbar-light" type="button" data-toggle="collapse" data-target="#main-navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="main-navigation">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="/">Home</a>
<li class="nav-item">
<a class="nav-link" href="/sources">Sources</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/about">About me</a>
</li>&nbsp;&nbsp;
<form action="{{ url_for('search') }}" method="post">
<input type="text" name="searchBar" placeholder="Search news here">
<button>Search</button>
</form>
</li>
</ul>
</div>
</nav>
<div class="card">

<div class="card-body">
<h5 class="card-title">404 Not Found</h5>
<p class="card-text">You may entered wrong url manually. Click below button to redirect to</p>

```

```
homepage</p>
<a href="/" class="btn btn-primary">Go to homepage</a> </div>
</div>
</body>
</html>
```

### main.py

```
from flask import Flask, render_template, url_for, send_from_directory, request from requests
import get
from os.path import join
app = Flask(__name__)
@app.route('/') def index():
#print(request.remote_addr)
response1 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'business'}).json()
response2 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'entertainment'}).json()
response3 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'general'}).json()
response4 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'health'}).json()
response5 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'science'}).json()
response6 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'sports'}).json()
response7 = get("https://newsapi.org/v2/top-headlines?country=in", params = {'apiKey' : 'Your
API Key',
'category' : 'technology'}).json()
return render_template('index.html', response1 = response1, response2 = response2, response3 =
response3, response4 = response4, response5 = response5, response6 = response6, response7 =
```

```
response7)
@app.route('/search', methods=['POST']) def search():
response = get("https://newsapi.org/v2/everything", params={'apikey' : 'Your API Key', 'q' :
request.form['searchBar']}).json()
return render_template('search.html', response = response)
@app.route('/sources')
def sources():
response = get("https://newsapi.org/v2/sources", params = {'apiKey' : 'Your API
Key'}).json()
return render_template('sources.html', name = response)
@app.route('/about') def about():
return render_template('about.html')
@app.errorhandler(404) def page_not_found(e):
return render_template('404.html'), 404
if __name__ == "__main__": app.run()
```

## 7.2 Server Side

### app.py

```
from flask import Flask, request, Response, send_from_directory
import requests
import os
import random
import json
from flask_cors import CORS
import db_crud

app = Flask(__name__) #, static_folder='build', template_folder='build', static_url_path="")
CORS(app)

NEWS_API_TOPHEADLINES_ENDPOINT = 'https://newsapi.org/v2/top-headlines'
NEWS_API_KEY = 'aeb9762b06d74d1a8ece0f3b896feb4c'

user_preferences = ['business','entertainment','general','health','science','sports','technology']

@app.route('/get-top-headlines', methods=['GET'])

def get_top_headlines_for_user():

    query = request.args.to_dict()
    if query.get('userName', -1) == -1:
        return Response('userName must be provided in query string', status=400)
    country = query.get('country', 'in')
    q = query.get('q', '')
    user_topics = db_crud.get_topics(query.get('userName'))
    user_prefs = 'category=' + '&category=' .join(user_topics)
    url =
    NEWS_API_TOPHEADLINES_ENDPOINT+f'?country={country}&{user_prefs}&apiKey={N
EWS_API_KEY}'

    if q != "": url += f'&q={q}'

    print(url)
```

```

api_response = requests.get(url=url)
return json.loads(api_response.content.decode()), 200

@app.route('/authenticate', methods=['POST'])
def authenticate_user():
    form = json.loads(request.data.decode())
    if db_crud.authenticate(form['email'], form['password']):
        return {'status':'success'}, 200
    return {'status':'failure'}, 400

@app.route('/sign-up', methods=['POST'])
def sign_up_user():
    form = json.loads(request.data.decode())
    print(form)
    if db_crud.create_account(form['name'], form['email'], form['password'], form['phone'],
form['dob'], [x['name'] for x in form['options']]):
        return {'status':'success'}, 200
    return {'status':'failure'}, 400

@app.route('/profile', methods=['GET'])
def user_profile():
    query = request.args.to_dict()
    ret = {
        "likes" : db_crud.get_likes(query['user_name']),
        "bookmarks" : db_crud.get_bookmarks(query['user_name']),
        "topics": db_crud.get_topics(query['user_name']),
        "user" : db_crud.get_user(query['user_name'])
    }
    return ret, 200

@app.route('/action', methods=["POST"])
def action():
    form = request.data.decode()

```

```
form = json.loads(form)
print(form)
if(form['type']=='A'):
    print("ADD")
    if db_crud.add_action(form['email'], form['title'], form['url'], form['urlToImage'],
form['publishedAt'], form['description'], form['action']):
        return {'status':'success'}, 200
    else:
        return {'status':'failure'}, 400
if(form['type']=='R'):
    print("REMOVE")
    if db_crud.remove_action(form['email'], form['url']):
        return {'status':'success'}, 200
    else:
        return {'status':'failure'}, 400
@app.route('/update-profile', methods=['POST'])
def updateprofile():
    form = json.loads(request.data.decode())
    db_crud.update_topics(form['user'], form['topics'])
    return {'status':'success'}, 200
# @app.route('/', methods=['GET'])
# def react_app():
#     return send_from_directory(app.static_folder, 'index.html')
# @app.errorhandler(404)
# def not_found(e):
#     return app.send_static_file('index.html')
if __name__ == '__main__':
    app.run(debug=True)
```

## dbCrud.py

```
import ibm_db
import os

dsn_hostname = "0c77d6f2-5da9-48a9-81f8-
86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
dsn_uid = "bsj69971"      # e.g. "abc12345"
dsn_pwd = "J0ww1ELveoQeDVWK"    # e.g. "7dBZ3wWt9XN6$o0J"
dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"        # e.g. "BLUDB"
dsn_port = "31198"          # e.g. "32733"
dsn_protocol = "TCPIP"       # i.e. "TCPIP"
dsn_security = "SSL"         #i.e. "SSL"

dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol,
                           dsn_uid, dsn_pwd, dsn_security)

conn = ibm_db.pconnect(dsn, "", "")

def authenticate(username, password):
    stmt = ibm_db.exec_immediate(conn, "SELECT * FROM USERS")
    row = True
    while row != False:
        row = ibm_db.fetch_assoc(stmt)
        if not row: return False
```

```

if(row['USERNAME'] == username and row['PASSWORD'] == password): return True
return False

def get_user(username):
    stmt = ibm_db.exec_immediate(conn, "SELECT * FROM USERS")
    row = True
    while row!= False:
        row = ibm_db.fetch_assoc(stmt)
        if not row: return False
        if(row['USERNAME'] == username):
            return dict(row)
    return dict()

def create_account(name, username, password, phone, dob, topics):
    try:
        sql = "INSERT INTO USERS VALUES(?,?,?,?,?,?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.bind_param(stmt, 2, username)
        ibm_db.bind_param(stmt, 3, password)
        ibm_db.bind_param(stmt, 4, phone)
        ibm_db.bind_param(stmt, 5, dob)
        ibm_db.bind_param(stmt, 6, '!'.join(topics))
        ibm_db.execute(stmt)
    return True
    except:
        return False

def get_likes(user):
    stmt = ibm_db.exec_immediate(conn, "SELECT * FROM NEWS N where N.user='{0}' and news_type='S'".format(user))
    likes = []

```

```

row = True
while row!= False:
    row = ibm_db.fetch_assoc(stmt)
    if not row: break
    likes.append(dict(row))
return likes

def get_bookmarks(user):
    stmt = ibm_db.exec_immediate(conn, "SELECT * FROM NEWS N where N.user='{0}' and news_type='B'".format(user))
    bookmarks = []
    row = True
    while row!= False:
        row = ibm_db.fetch_assoc(stmt)
        if not row: break
        bookmarks.append(dict(row))
    return bookmarks

def get_topics(user):
    stmt = ibm_db.exec_immediate(conn, "SELECT * FROM USERS where username='{0}'".format(user))
    row = True
    while row!= False:
        row = ibm_db.fetch_assoc(stmt)
        if not row: break
    return row['TOPICS'].split(',')

def update_topics(user, topics):
    sql = "UPDATE USERS SET topics=? where username=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1,''.join(topics))

```

```
ibm_db.bind_param(stmt, 2, user)
ibm_db.execute(stmt)

def add_action(user, title, url, image_url, date, desc, action):
    try:
        sql = "INSERT INTO NEWS VALUES(?,?,?,?,?,?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, user)
        ibm_db.bind_param(stmt, 2, title)
        ibm_db.bind_param(stmt, 3, url)
        ibm_db.bind_param(stmt, 4, image_url)
        ibm_db.bind_param(stmt, 5, date)
        ibm_db.bind_param(stmt, 6, desc)
        ibm_db.bind_param(stmt, 7, action)
        ibm_db.execute(stmt)
    return True
except Exception as e:
    print(e)
    return False

def remove_action(user, url):
    try:
        sql = "DELETE FROM NEWS N WHERE N.user='{0}' AND
news_article_link='{1}'".format(user, url)
        ibm_db.exec_immediate(conn, sql)
    return True
except Exception as e:
    print(e)
    return False
```

### 7.3 Schema

```
CREATE TABLE USERS(
    name varchar(255),
    username varchar(255) not null primary key,
    password varchar(255),
    phno varchar(10),
    dob varchar(10),
    topics varchar(10000)
);
```

```
CREATE TABLE NEWS(
    user varchar(255),
    news_title varchar(255),
    news_article_link varchar(1024),
    news_image_link varchar(1024),
    news_date varchar(255),
    news_description varchar(255),
    news_type varchar(1) check (news_type in ('B','L')) ,
    foreign key(user) references USERS(username)
);
```

# TESTING

## 8.1 Acceptance Testing

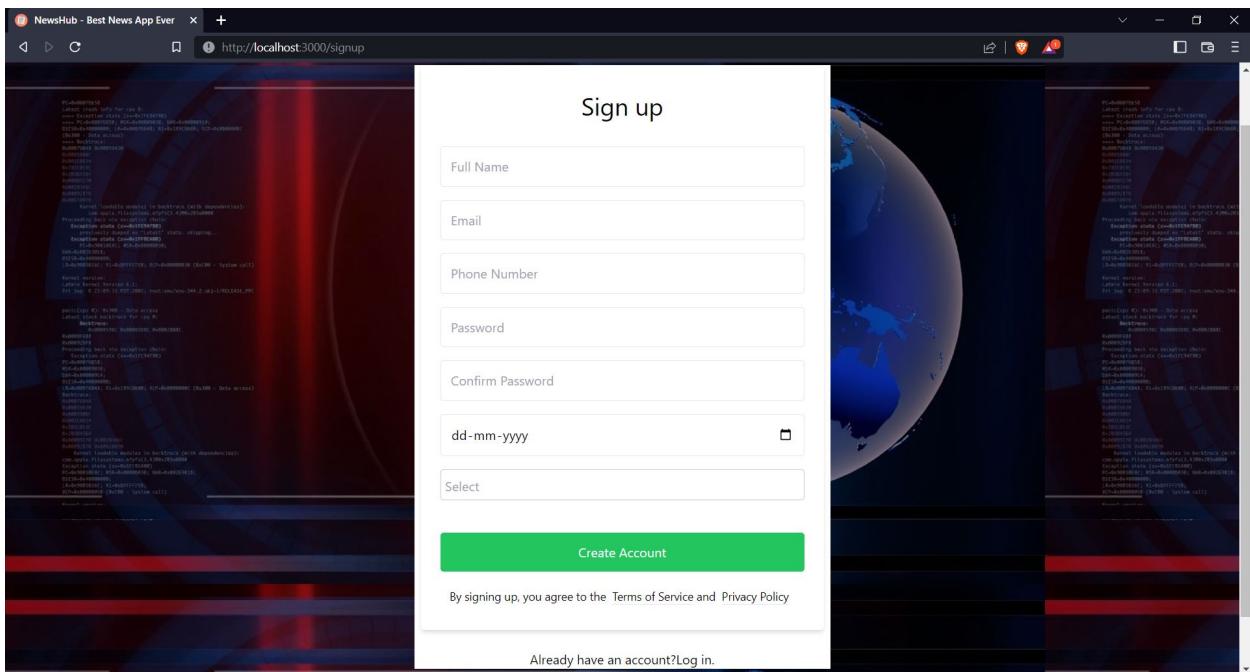
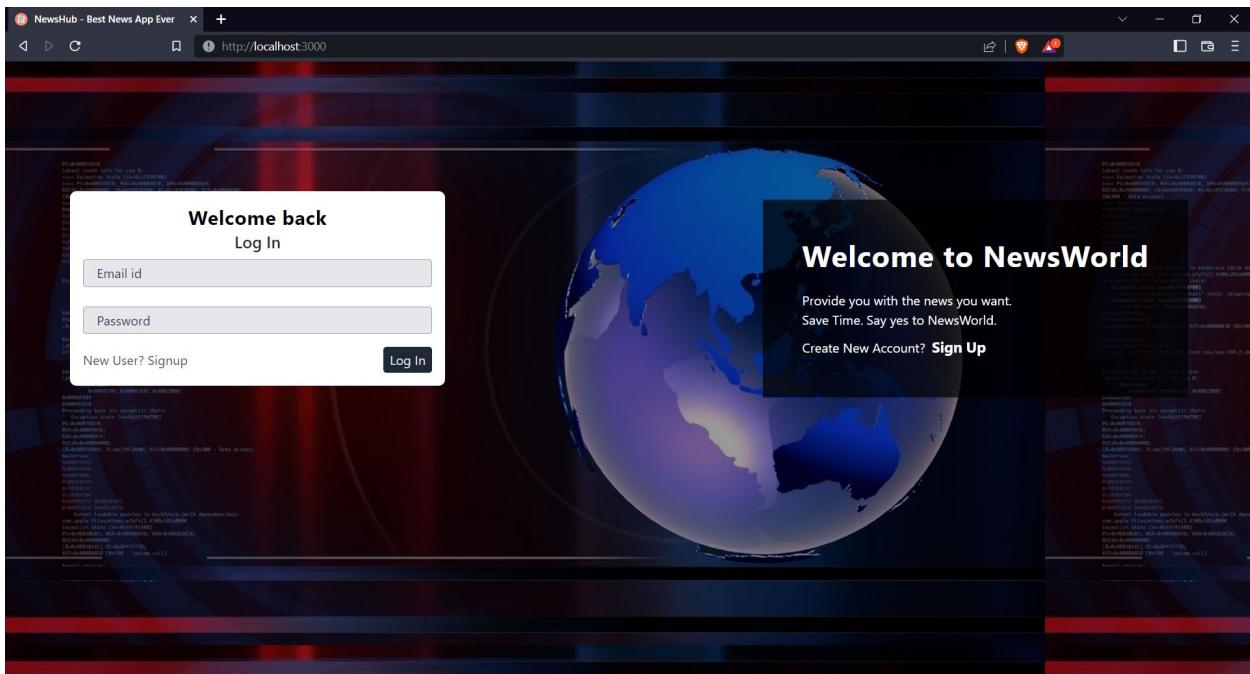
Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	BUG ID	Executed By
Cloud Account	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.select OS installation disk space main memory space.	<a href="https://shopenzer.com/">https://shopenzer.com/</a>	Login/Signup popup should display	Working as expected	Pass			Sabari Rohith
Docker Cli	UI	Home Page	Storing web pages(web app)	1. Create file system docker,2.store web page	<a href="https://shopenzer.com/">https://shopenzer.com/</a>	Application should show below UI elements: a.email text box b.password text box	Working as expected	Fail	Steps are not clear to follow	BUG-1 234	Sabareesan
Create send grid	Functional	Home page	Verify user is able to log into application with Valid credentials	1.create account to integrate application to DBI server API	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage		Pass			Kavinayaa
Create IBM DB2	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Create personal details scheme DB2,2.Create employment vacancy scheme in DB2	Username: chalam@gmail.com password: Testing123	Application should show 'Incorrect email or password' validation message.		Pass			Dharanya
Connect frontend to send grid DB2	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Web url address of DB2 personal details DB2 and employer DB2,2.Web API address has to be integrated	Username: chalam@gmail.com password: Testing12367868678687687	Application should show 'Incorrect email or password' validation message.		Pass			Kavinayaa
Creating Cluster	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Design the HTML web page,2.Docker image of HTML webpage is installed in container 3.web API is created in web server	Username: chalam password: Testing123678686786876876	Application should show 'Incorrect email or password' validation message.		Pass			Sreeram
Personal detail collection	functional	admin page	submitting the required details		1.user intersted categories news,2.submitting the web page stored in DB2.			Pass			Sabareesan
Employee details gathering	functional	user details page	submitting the user details		2.news description,3.web page respond stored in DB2.			Pass			Sabari Rohith
Job search	functional	Registration -User, AdminLogin -User,		1.Enter news categories,share the news in social media ,2.Query the DB2,3.Suitable responses are displayed				Pass			Dharanya

## 8.2 Test Cases Report

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Bug ID	Executed By
LoginPage_TID_01	Functional	Home Page	Verify user is able to see the Login/Signup p when user clicked on account LOGIN button	1.Enter URL and click go 2.Click on Account Login dropdown button 3.Verify login/Singup popup displayed or not	<a href="https://shopenzer.com/">https://shopenzer.com/</a>	Login/Signup popup should display	Working as expected	Pass			Sabareesan
LoginPage_TID_02	UI	Home Page	Verify the UI elements in Login/Signup popup	1.Enter URL and click go 2.Click on Account Login dropdown button 3.Verify login/Singup with below UI elements: a.email box b.password box c.Login button d.New customer? Create e.Register link e.Last password? Recovery password link	<a href="https://shopenzer.com/">https://shopenzer.com/</a>	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-1 234	Kavinaya
LoginPage_TID_03	Functional	Home page	Verify user is log into application with Valid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on Account Login dropdown button 3.Enter Valid username/email in Email 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					Sreeram
LoginPage_TID_04	Functional	Login page	Verify user is log into application with InValid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on Account Login 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.com password: Testing123	Application should show 'Incorrect email or password' validation message.					SabariRohith
LoginPage_TID_05	Functional	Login page	Verify user is log into application with InValid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on Account Login 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: Testing12367868678687687687 6	Application should show 'Incorrect email or password' validation message.					Dharanya
LoginPage_TID_06	Functional	Login page	Verify user is log into application with InValid credentials	1.Enter URL( <a href="https://shopenzer.com/">https://shopenzer.com/</a> ) and click go 2.Click on Account Login dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam password: Testing12367868678687687687 6	Application should show 'Incorrect email or password' validation message.					Sabareesan

# RESULTS

## 9.1 Sample Screens



NewsHub - Best News App Ever

BTS's Jungkook drops cover image for his 'FIFA World Cup Qatar 2022' soundtrack, 'Dreamers' - allkpop

Sat Nov 19 2022 10:56:00

BTS's Jungkook has dropped an epic cover image for his official soundtrack for the upcoming 'FIFA World Cup Qatar 2022'! "Dr...

1 Likes 1 Bookmarks

LOGOUT

NewsHub - Best News App Ever

http://localhost:3000/profile

1 Likes 1 Bookmarks

**Sab**

DATE OF BIRTH : 2005-07-13

Email : sab@gmail.com

Password : \*\*\*\*\*

Topics: entertainment,general,science,business,Virat Kohli

---

1 of 1

**Articles Liked**

When Zeenat Aman became the first Indian to win Miss Asia Pacific title at 19 - Hindustan Times

Sat Nov 19 2022 06:44:44

1 of 1

NewsHub - Best News App Ever

http://localhost:3000/profile

1899  
A NETFLIX SERIES  
NOVEMBER 17 | NETFLIX

DARK  
A NETFLIX ORIGINAL SERIES  
ALL EPISODES DECEMBER 1  
NETFLIX

1899 on Netflix is an engaging watch but if you start watching it expecting it to be like Dark, which is one of the best International series on Netflix, you will be sorely disappointed.

• • •

### Update Tags

entertainment general science business Virat Kohli Select

UPDATE

## **ADVANTAGES AND DISADVANTAGES**

### **10.1 Advantages**

1. All trending news will be displayed in a orderly manner.
2. We can customize the topic which we are interested and make them display
3. The user experience are quit simple and friendly

### **10.2 Disadvantages**

1. Same news which is published in different websites can be displayed.
2. Delay response of news api can make the news collapse.

## **FUTURE WORKS**

1. Location feature with automation can be implemented which means as user move from one city to other local news will change as per it.
2. Offline Reading can be improve will more efficient way on full articles.
3. Data quality check needed.
4. If API can't reach to certain article source it gives null value which can cause problem in JSON parsing.

## **REFERENCES**

1. Sangeeta Ruth, Srividhya Raghavan V, Smrithi J, Saira Banu. 2016. “Spatial Preference Newsfeed System For Android Mobile Users”, IJCSITS, Vol6, NO. 3: 24.
2. <https://newsapi.org/>
3. <https://dzone.com/articles/how-to-parse-json-datafrom-a-rest-api-using-simpl>
4. <https://material.io/>
5. <https://developer.android.com/guide>