News Tracker Application

Submitted by

PNT2022TMID04885

Thamarikanna G-737819ITR10

Vani Sridhar C-737819ITR102

Varshini P-737819ITR103

Vasanth T-737819ITR104

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Flask is a python framework provides the environment to create web application using python with the knowledge of HTML and CSS. We work with different modules. Application programming interface (API) is act as a interface between different applications.

Using IBM cloud make the application accessible from anywhere. In difficult times like corona or in remote places getting newspaper is difficult thing. So, this application makes it simple with the availability of network. Also is helps user to choose their interested domain and also allows them to search about the particular news.

1.2 PURPOSE

The purpose of this application are listed below:

- To help user to get news even in remote places or in lockdown period.
- Helps to reduce the cost required to read news. And encourage the user to read news.
- To provide the option for the user to choose their interested domain.
- To provide search option for the user to search for the needed information.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

There is a lot of time consumption to read the relevant news that we need to read without categorizing.with the help of categorizing we are able to read the news which we need.only read what need will save time.unable to follow some big publishers and their current news feed.

2.2 REFERENCE

1]https://www.ripublicatio n.com/ijaer18/ijaerv13n 11_78.pdf

2] (PDF) Exploring mobile news reading interactions for news app personalisation (researchgate.net)

3] (PDF) Topic Detection and Tracking in News Articles (researchgate.net)

4](PDF)Implementing a Mobile Application News Tool for Disseminating Messages and Events of AlBuraimi University College

5](PDF) Following the Fed with a News Tracker (researchgate.net)

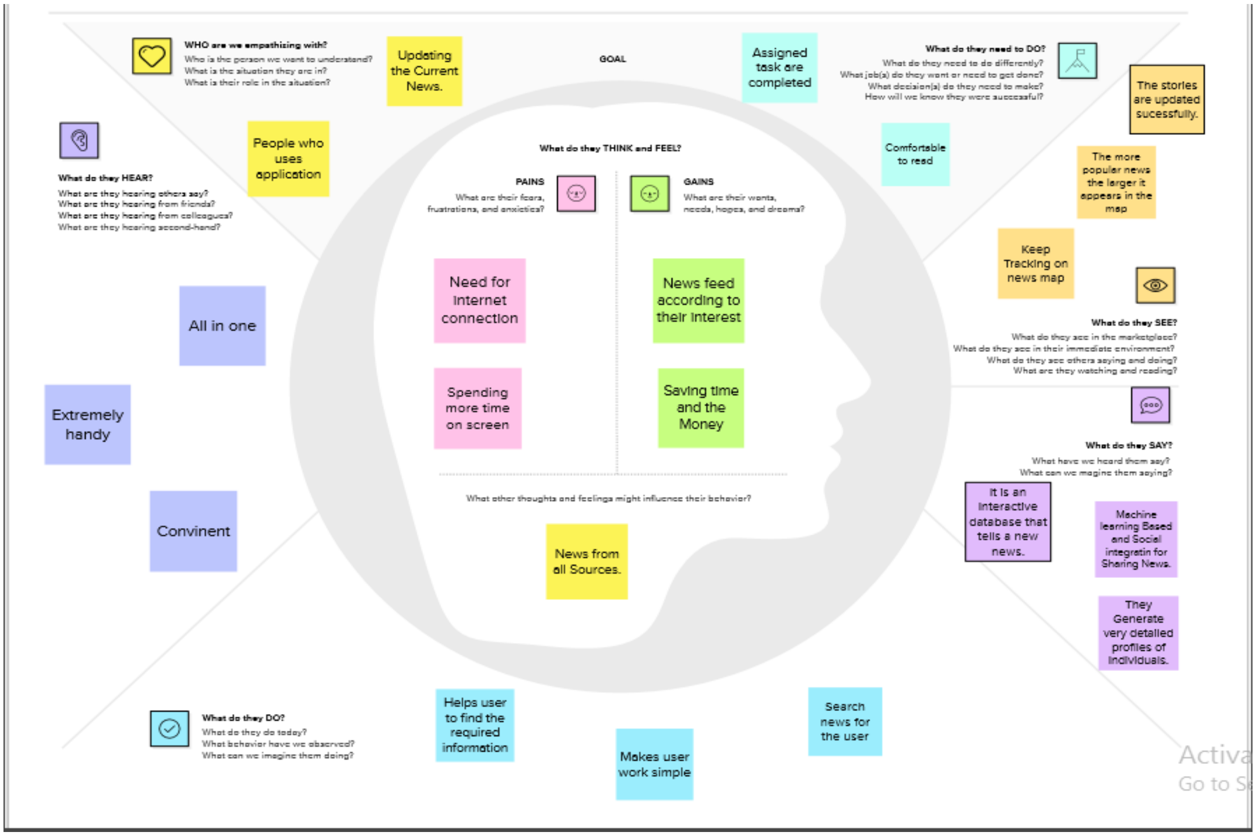
2.3 PROBLEM STATEMENT DEFINITION

News reading increases peoples knowledge. But now a days news reading habbit reduced due to increase of technology and not getting proper source for reading news in pandemic times. So this application helps the user to read the news anywhere the want and they no need to carry the newspaper everywhere. The application will be user friendly so it encourages the user to read news.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

Updating the Current News.

People who uses application

GOAL

Assigned task are completed

What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

The stories are updated sucessfully.

Comfortable to read

The more popular news the larger it appears in the map

Keep Tracking on news map

What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

All in one

Extremely handy

Convinent

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

Need for Internet connection

Spending more time on screen

GAINS
What are their wants, needs, hopes, and dreams?

News feed according to their interest

Saving time and the Money

What other thoughts and feelings might influence their behavior?

News from all Sources.

What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?
What have we heard them say?
What can we magine them saying?

It Is an Interactive database that tells a new news.

Machine learning Based and Social integratin for Sharing News.

They Generate very detailed profiles of Individuals.

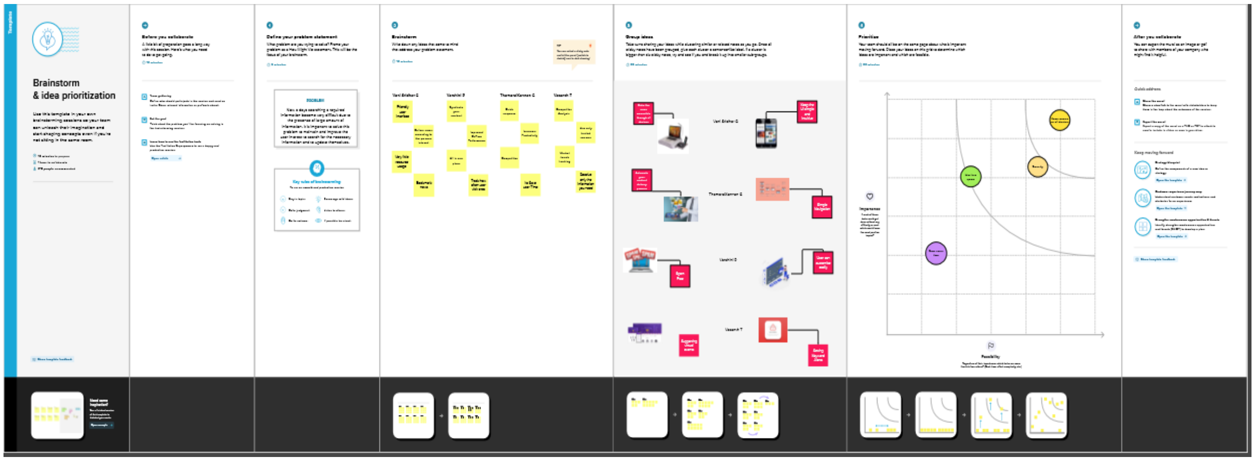
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

Helps user to find the required information

Makes user work simple

Search news for the user

3.2 IDEATION & BRAINSTORMING

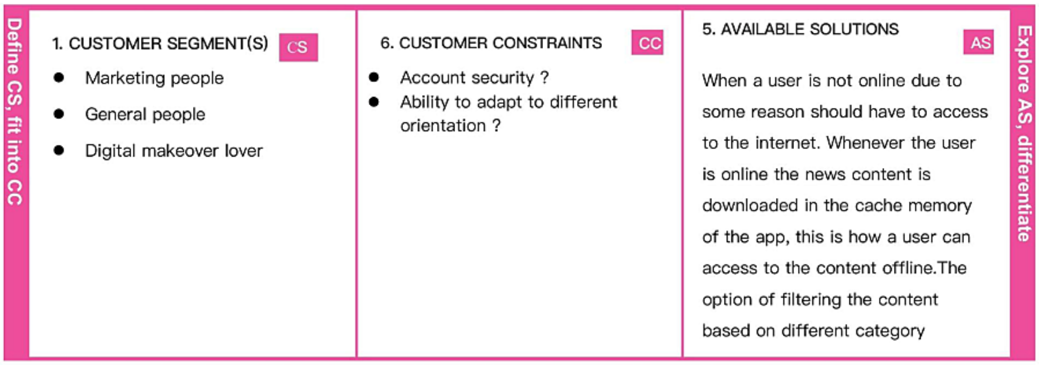


3.3 PROPOSED SOLUTION

S.No.	Proposals	Description
1	Problem Statement	Now a days searching a required information become very difficult due to the presence of large amount of information. It is important to solve this problem to maintain and improve the user interest to search for the necessary information and to update themselves. And also user needs to buy the newspaper or have to carry the newspaper everywhere.
2	Solution description	The application to be developed should be accessible from anywhere and any devices that are connected to the internet. The application should allow the user to search for the specific news that user needs to know.

3	Uniqueness	There are many applications that will provide unwanted messages along with required messages. Our work is develop a cloud application that can be accessed anywhere and it should allow the user to search for the information.
4	Social Impact	In remote areas or in critical times, it is not possible to read the newspaper. But we can able to use app instead.
5	Business Model	Right now the application is profitless but in future we might add an option of premium plans for advanced learning.
6	Scalability of the Solution	Compare to normal newspapers this app provide more information. And it provides news about other countries or other regions.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Marketing people
- General people
- Digital makeover lover

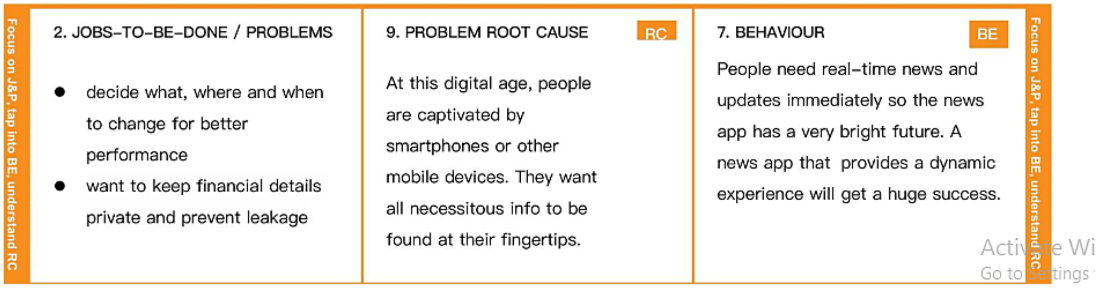
6. CUSTOMER CONSTRAINTS CC

- Account security ?
- Ability to adapt to different orientation ?

5. AVAILABLE SOLUTIONS AS

When a user is not online due to some reason 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.The option of filtering the content based on different category

Explore AS, differentiate



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

2. JOBS–TO–BE–DONE / PROBLEMS

- decide what, where and when to change for better performance
- want to keep financial details private and prevent leakage

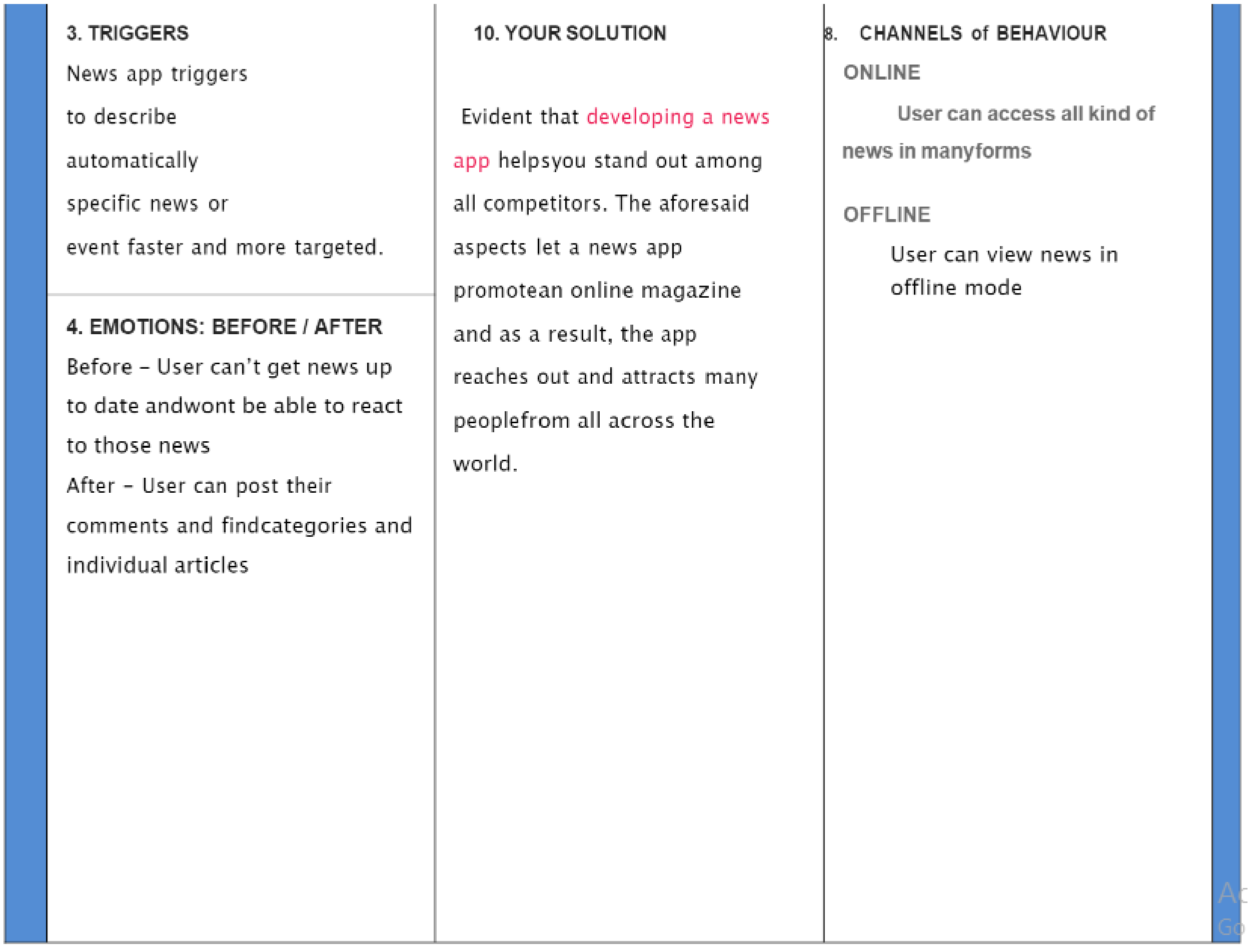
9. PROBLEM ROOT CAUSE RC

At this digital age, people are captivated by smartphones or other mobile devices. They want all necessitous info to be found at their fingertips.

7. BEHAVIOUR BE

People need real–time news and updates immediately so the news app has a very bright future. A news app that provides a dynamic experience will get a huge success.

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



3. TRIGGERS

News app triggers to describe automatically specific news or event faster and more targeted.

4. EMOTIONS: BEFORE / AFTER

Before – User can't get news up to date andwont be able to react to those news

After – User can post their comments and findcategories and individual articles

10. YOUR SOLUTION

Evident that developing a news app helpsyou stand out among all competitors. The aforesaid aspects let a news app promotean online magazine and as a result, the app reaches out and attracts many peoplefrom all across the world.

8. CHANNELS of BEHAVIOUR

ONLINE

User can access all kind of news in manyforms

OFFLINE

User can view news in offline mode

CHAPTER 4

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
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login through browser directly by entering username and password Login through app
FR-4	User interaction	Done through user interface between client and server View the related news by subscripted or requested page
FR-5	Tools	Application have tools to share this news in social networks

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability is a measure of how well a specific user in a specific context can use a product/design to achieve a defined goal effectively, efficiently and satisfactorily. Designers usually measure a design’s usability throughout the development process—from wireframes to the final deliverable—to ensure maximum usability.
NFR-2	Security	Assuring all data inside the system or its part will be protected against malware attacks or unauthorized access.
NFR-3	Reliability	The probability that a product, system, or service will perform its intended function adequately for a specified period of time, or will operate in a defined environment without failure. The most important components of this definition must be clearly understood to fully know how reliability in a product or service is established
NFR-4	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 operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment.

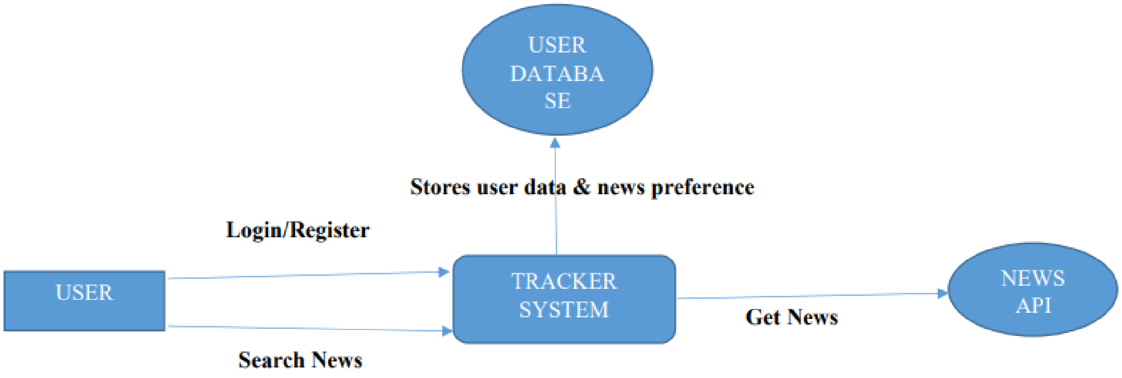
NFR-5	Availability	Availability describes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period. 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
NFR-6	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.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

A Data Flow Diagram is a traditional visual representation of the information flows within a system. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Technical Architecture:

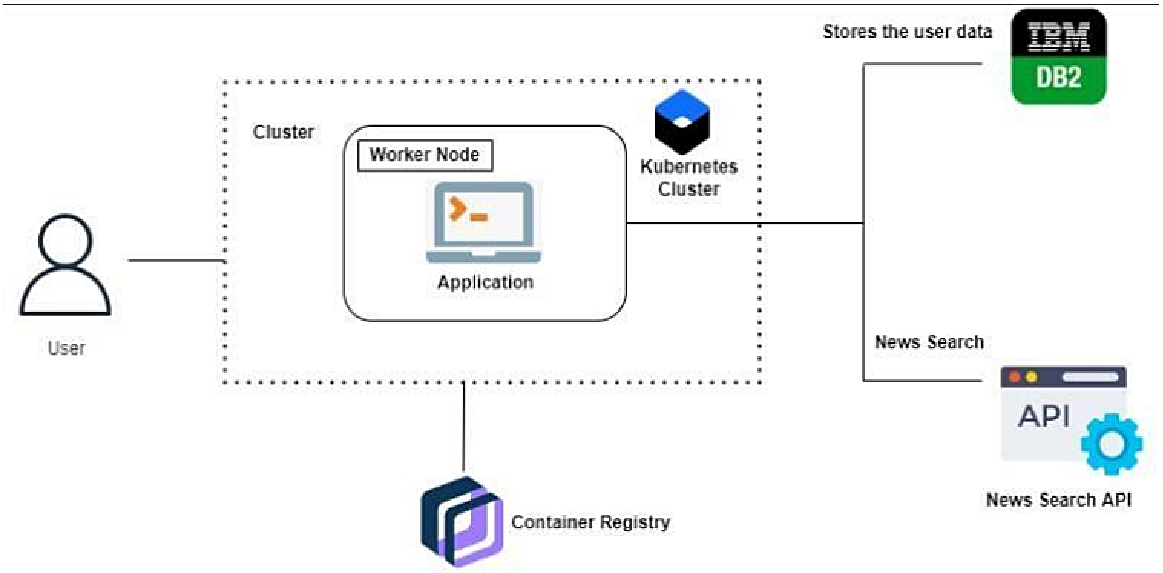


Table-1: Components and Technologies:

S. No	Component	Description	Technology
1	User Interface	How user interacts with application, Ex. Mobile App, Chatbot	HTML, CSS, Python, MySQL, Flask
2	Application Logic-1	Logic for a process in the application	Python
3	Application Logic-2	Logic for a process in the application	SendGrid

4	Application Logic-3	Logic for a process in the application	IBM Watson Application
5	Database	Data Type, Configuration etc.	MySQL
6	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud
7	File Storage	File storage requirements	IBM Object Storage

Table-2: Application characteristics:

S. No	Characteristics	Description	Technology
1	Open-source frameworks	Microweb framework written in Python	Flask
2	Security implementation	A set of cryptographic hash function	SHA-256
3	Scalable Architecture	Kubernetes is a open source container orchestration system and management	Kubernetes
4	Availability	To customize settings for the docker CLI. The configuration file uses JSON formatting.	Docker CLI
5	Performance	The performance will be high because there will be no network traffics in the application	Kubernetes is cluster

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user / Web User)	Registration	USN-1	User can register for the application by entering my email, password.	I can access my account / dashboard	High	Sprint-1
		USN-2	User will receive confirmation email once he/she have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	User can register for the application through Gmail		Low	Sprint-1
	Login	USN-4	User can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	User can enter the interests and choices of news that he want to see for the first		High	Sprint-2

			time in dashboard.			
		USN-6	User can go through the feed of news filtered according to their wish.		High	Sprint-3
		USN-7	User can log out of his account in settings.	I can click confirm to log out and end the session	High	Sprint-3
	Settings	USN-8	User can update their interests and choice in account settings.		Medium	Sprint -4
Customer Care Executive	Chat Bot / Query Section	USN-9	Solve issues brought up by client		High	Sprint -4
Administrator		USN-10	Roll out updates and bug fix		High	Sprint -4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint-1:

Registration:

The user can register to by using their username and password. It will be stored in IBM database.

MySQL Database Connection :

MySQL database is connected to the application. the database has table "user". It has id, username and password.

categories:

The user can search for a perticular news that they need to read using search option.

Login:

User can login to application by using the password and eamil.

Sprint-2

Dashboard:

User can select the news category that they want to view in the dashboard. And can search the news they needed.

Dashboard User Interface:

Administrator designing the user interface.

Sprint-3

IBM cloud Connection:

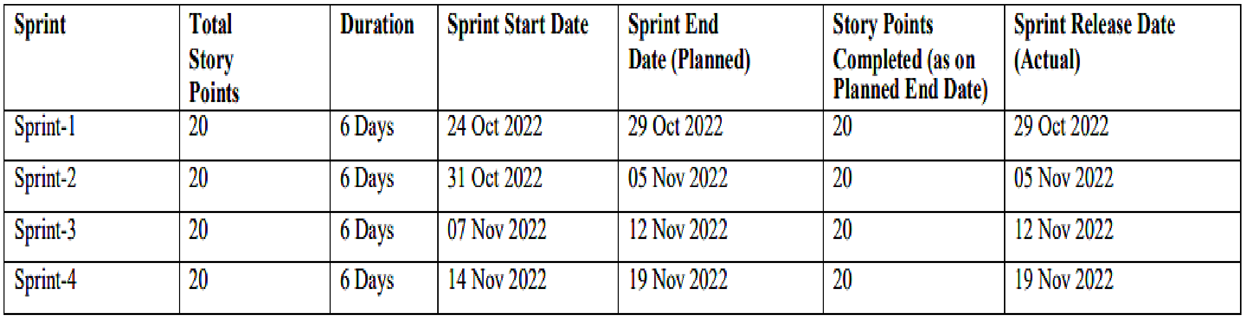
Our application is connected to the cloud.

Sprint-4

Chat bot / Query:

The user can know the information about our application using bot.

6.2 SPRINT DELIVERY SCHEDULE



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

CHAPTER 7
CODING & SOLUTIONING

7.1 FEATURE 1

SEARCH

It allows user to search for the specfic information that they need to know.

API:

```
news_data =
requests.get(f'https://newsapi.org/v2/everything?q={arg3}&apiKey={NEWS_API_KEY}
').json()
return news_data['articles']
```

Python file:

```
#search
@app.route('/search', methods = ['POST', 'GET'])
def search():
    if request.method == 'POST':
        if 'status' in session:
            sr = request.form['search']
            news_articles = find(sr)
            return render_template("news.html", news_articles=news_articles)
        else:
            msg = "Please Login!!"
            return render_template("result.html", msg = msg)
```

7.2 FEATURE 2

IBM Chatbot

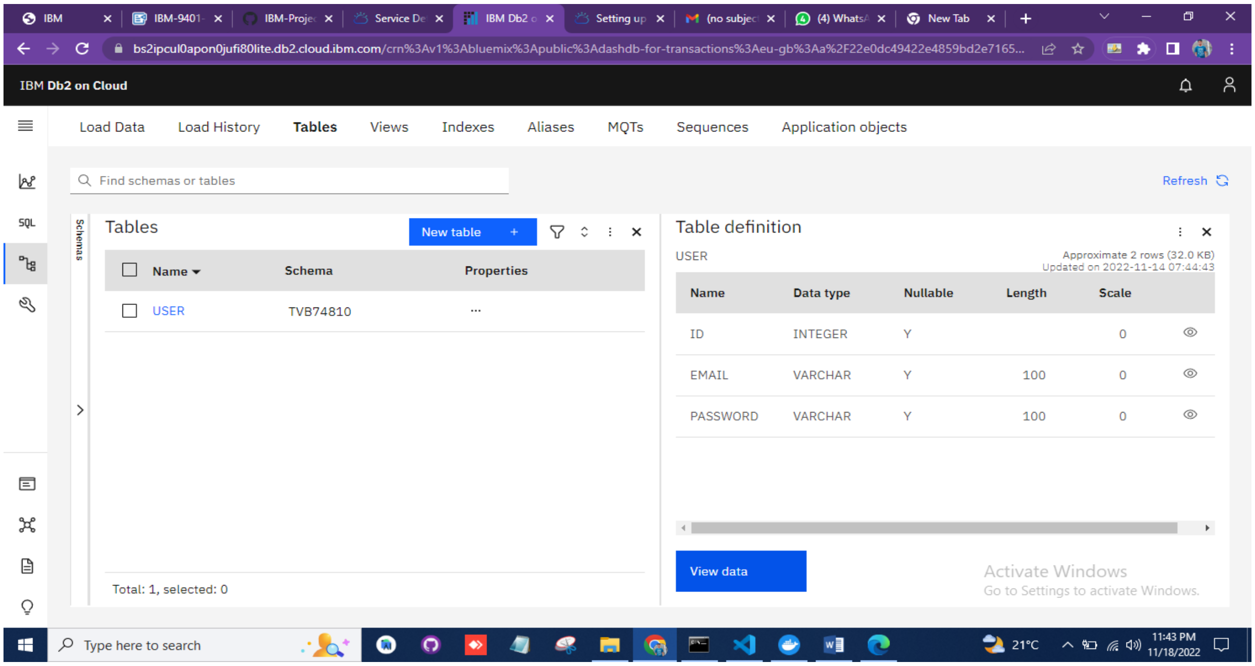
User can interact with chatbot and can get the information about the features of the application.

CODE:

```
<script>
    window.watsonAssistantChatOptions = {
      integrationID: "380464dc-d472-4156-bdf1-3ba4aca9e386", // The ID of this
integration.
      region: "jp-tok", // The region your integration is hosted in.
      serviceInstanceID: "5c2a539f-09e4-4815-bfe8-33e934287400", // The ID of your
service instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const t=document.createElement('script');
      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
      document.head.appendChild(t);
    });
   </script>
```

7.3 DATABASE SCHEMA

The database has table "user". The user has three columns Id, Email and Password.



Name	Schema	Properties
USER	TVB74810	···

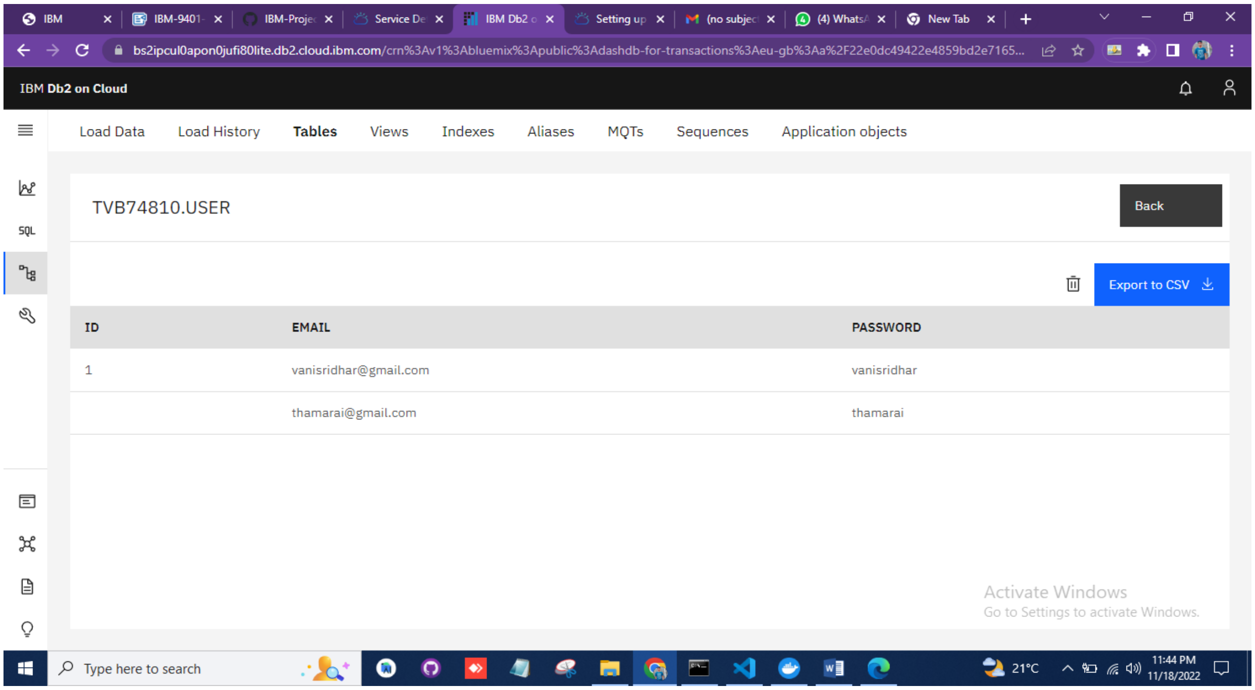
Total: 1, selected: 0

Table definition

USER

Approximate 2 rows (32.0 KB)
Updated on 2022-11-14 07:44:43

Name	Data type	Nullable	Length	Scale
ID	INTEGER	Y		0
EMAIL	VARCHAR	Y	100	0
PASSWORD	VARCHAR	Y	100	0



TVB74810.USER

ID	EMAIL	PASSWORD
1	vanisridhar@gmail.com	vanisridhar
	thamarai@gmail.com	thamarai

CHAPTER 8

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

The advantages of this application are:

- We can assess this application from anywhere.
- The user can get the news even in lockdown situations.
- The user can get the news even in remote areas.
- It reduces the cost of reading newspaper, so it helps in encouraging the user to read news.
- The user can choose the category which the wise to read.
- This application provides the search option, so that user can search the needed news.
- No need to download this application.

DISADVANTAGES

The disadvantage of this application are:

- There should be proper internet connectivity.
- The user should have smart phone or laptop.
- Reduced speed compare to the application that is hosted on a local server.
- Restricted Functionality.

CHAPTER 9

CONCLUSION

Hence this application satisfies the user and encourage them to study the news. They can able to read the news from compact mobile phones to so they no need to carry the newspapers to every place they go. With good interface the users are satisfied. By collecting the requirements, the users are provided the news categories that they can choose to need. And also the search option is provided for user to search for the particular news they need.

This application is reliable with good user interface. The application is secure with the Secure Sockets Layer. The use of IBM cloud makes the application available in all places including remote areas with the presence of good internet facility.

There is no cost for reading this application and user no need to download the application. It is very user friendly and it is like mobile newspaper (i.e it is very compact to use).

CHAPTER 10

FUTURE SCOPE

Providing the facility to the user to choose the country the need to know. Locating the location of the user and providing the news based on his/her location. And changing of news data if the users moves from one location to other. And providing the facility to the user to read the news offline. The security of the user data can be improved. The user can be verified before registering.

CHAPTER 11

APPENDIX

SOURCE CODE:

Home.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">
   <title>Home</title>

   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>

   <script>
      function getnews(){
         fetch("https://newsapi.org/v2/top-
headlines?category=general&apiKey=cdc40c3ada3e44b9b096688f732b4116")
         .then(a=>a.json())
```

```
            .then(response =>{
                for(var i=0; i<response.articles.length; i++){
                        document.getElementById("output").innerHTML+="<div class='table-responsive' style='color:white;'><table class='table table-sm table-hover table-striped table-dark'><tbody style='font-size: 15px;'><tr><td><img style='width:300px;height:200px;border-style:solid;' src='"+response.articles[i].urlToImage+"'></td><td><h3 style='font-size: 20px;'>"+response.articles[i].title+"</h3><h4 style='font-size: 18px;color: lime;'>"+response.articles[i].source.name+"</h4><br>"+response.articles[i].description+" <a href='"+response.articles[i].url+"'
target='_blank'>"+response.articles[i].url+"</a></td></tr></tbody></div>";
                }
            })
            }
    </script>
</head>
<body onload="getnews()" style="background-color: rgb(2, 1, 12);">

    <div class="container-fluid" style="background: linear-gradient(to bottom, #000099  0%, #003333 100%);">
    <div class="row">
        <div class="col-md-12 mt-3 mb-3">
            <h2 class="text-center" style="color: white">Latest News Headlines</h2>
        </div>
    </div>
  </div>

          <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd; margin-bottom: 15px;">
        <a class="navbar-brand" href="/" style="color: red; font-weight: bold;">News App</a>
                  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav mr-auto">
```

```
      <li class="nav-item active">
       <a class="nav-link active" href="home" style="font-weight: 500;">Home</a>
      </li>
      <li class="nav-item dropdown">
       <a class="nav-link dropdown-toggle active" style="font-weight: 500;" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Category
       </a>
       <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="entertainment">Entertainment</a>
        <a class="dropdown-item" href="sports">Sports</a>
        <a class="dropdown-item" href="technology">Technology</a>
        <a class="dropdown-item" href="science">Science</a>
        <a class="dropdown-item" href="health">Health</a>
        <a class="dropdown-item" href="business">Business</a>
       </div>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="/logout" style="color: black;font-weight: 500;">Logout <span class="sr-only">(current)</span></a>
      </li>
     </ul>
     <form class="form-inline my-2 my-lg-0" action="/search" method = "POST">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search" name="search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
     </form>
    </div>
   </nav>

  <div id="output"></div>
</body>
</html>
```

Index.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">
    <title>Welome</title>

                                        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>

    <style>
        body{
                        background-image: url('https://cvs-images.s3.jp-tok.cloud-object-
storage.appdomain.cloud/bg8.jpg');
            background-position: center center;
            background-repeat: no-repeat;
            background-attachment: fixed;
            background-size: cover;
            background-color: #000033;
        }
```

```
  </style>
</head>
<body>
     <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd; margin-bottom: 15px;">
    <a class="navbar-brand" href="/" style="color: red; font-weight: bold;">News App</a>
             <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
     <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
     <ul class="navbar-nav mr-auto">
      <li class="nav-item">
         <a class="nav-link" href="/login" style="color: black;font-weight: 500;">Login <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
                    <a class="nav-link" href="/register" style="color: black;font-weight: 500;">Register</a>
      </li>
     </ul>
    </div>
   </nav>

   <div class="container-fluid">
    <div class="row">
      <div class="col-md-12 mt-3 mb-3">
           <h1 class="text-center" style="color: white; margin-top: 20%;">Welcom To News App!!</h1>
      </div>
    </div>
   </div>

<script>
   window.watsonAssistantChatOptions = {
        integrationID: "380464dc-d472-4156-bdf1-3ba4aca9e386", // The ID of this integration.
        region: "jp-tok", // The region your integration is hosted in.
          serviceInstanceID: "5c2a539f-09e4-4815-bfe8-33e934287400", // The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    </script>

</body>
</html>
```

News.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <!-- Bootstrap CSS 4-->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
```

```
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>

  <!-- Bootstrap CSS 5-->
  <link
   href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
   rel="stylesheet"
   integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
   crossorigin="anonymous"
  />

  <title>Latest News Headlines</title>
 </head>
 <body>
  <div class="container-fluid" style="background-color: #00a3a3;">
   <div class="row">
     <div class="col-md-12 mt-3 mb-3">
       <h2 class="text-center" style="color: white">{{ cat }}</h2>
     </div>
   </div>
  </div>
  <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd;
margin-bottom: 15px;">
   <a class="navbar-brand" href="/" style="color: red; font-weight: bold;">News App</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
```

```
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link active" href="home" style="font-weight: 500;">Home</a>
          </li>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle active" style="font-weight: 500;" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
              Category
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdown">
              <a class="dropdown-item" href="entertainment">Entertainment</a>
              <a class="dropdown-item" href="sports">Sports</a>
              <a class="dropdown-item" href="technology">Technology</a>
              <a class="dropdown-item" href="science">Science</a>
              <a class="dropdown-item" href="health">Health</a>
              <a class="dropdown-item" href="business">Business</a>
            </div>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/logout" style="color: black;font-weight: 500;">Logout <span class="sr-only">(current)</span></a>
          </li>
        </ul>
        <form class="form-inline my-2 my-lg-0" action="/search" method = "POST">
          <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search" name="search">
          <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
        </form>
      </div>
    </nav>

    <div class="row mt-5">
      <div class="col-md-2"></div>
      <div class="col-md-8">
```

```
        <div class="row">
          {% for news in news_articles %}
          <div class="col-md-4 mb-5">
            <div class="card" style="width: 18rem">
              <img src="{{ news.urlToImage }}" class="card-img-top" alt="..." />
              <div class="card-body">
                <h5 class="card-title">{{ news.title }}</h5>
                <p class="card-text">{{ news.description }}</p>
                <a href="{{ news.url }}" class="btn btn-primary">Read More</a>
              </div>
            </div>
          </div>
          {% endfor %}
        </div>
      </div>
      <div class="col-md-2"></div>
    </div>
    <!-- Bootstrap Bundle with Popper -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
      crossorigin="anonymous"
    ></script>

    <script>
      window.watsonAssistantChatOptions = {
        integrationID: "380464dc-d472-4156-bdf1-3ba4aca9e386", // The ID of this integration.
        region: "jp-tok", // The region your integration is hosted in.
        serviceInstanceID: "5c2a539f-09e4-4815-bfe8-33e934287400", // The ID of your service
instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
```

```
        document.head.appendChild(t);
      });
    </script>

 <script>
      window.watsonAssistantChatOptions = {
        integrationID: "380464dc-d472-4156-bdf1-3ba4aca9e386", // The ID of this integration.
        region: "jp-tok", // The region your integration is hosted in.
        serviceInstanceID: "5c2a539f-09e4-4815-bfe8-33e934287400", // The ID of your service
instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    </script>

  </body>
</html>
```

Register.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">
    <title>Register</title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
```

```
crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>

  <style>
    h3{
      color: antiquewhite;
    }
    #form{
      margin: 10%;
      width: 30%;
      height:fit-content;
      color:antiquewhite;
      background-color: #202020;
      border-style: solid;
      border-radius: 10px;
      padding: 10px;
      text-align: left;
      opacity: 0.9;
    }

    #ele{
      padding: 10px;
      color: rgb(250, 248, 245);
      font-weight: bold;
    }

    #button1{
        font-weight: bold;
        color: #fbfbfd;
    }

    #button1:hover{
        color: #000033;
    }

    h3{
        color: antiquewhite;
    }

    body{
        background-image: url('static/bg2.jpg');
        background-position: center center;
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;
        background-color: #000033;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd;
margin-bottom: 15px;">
    <a class="navbar-brand" href="/" style="color: red; font-weight: bold;">News App</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
      <li class="nav-item">
          <a class="nav-link" href="/login" style="color: black;font-weight: 500;">Login <span
class="sr-only">(current)</span></a>
      </li>
            <li class="nav-item">
              <a class="nav-link" href="/register" style="color: black;font-weight:
500;">Register</a>
            </li>
          </ul>
        </div>
      </nav>
    <center>
      <form id="form" action="/reg_res" method = "POST">
        <div class="form-group">
          <label for="exampleInputName1" id="ele">Email Address</label>
          <input type="email" class="form-control" id="exampleInputName1" name="email"
placeholder="Enter Your Email">
          <small id="emailHelp" class="form-text" style="font-weight: 600;">We'll never share
your email with anyone else.</small>
        </div>
        <div class="form-group">
          <label for="exampleInputEmail1" id="ele">Password</label>
          <input type="password" class="form-control" id="exampleInputEmail1" name =
"password" aria-describedby="emailHelp" placeholder="Enter Your Password">
        </div>
        <center><button type="submit" class="btn btn-outline-light"
id="button1">Submit</button></center>
      </form>
    </center>
</body>
</html>
```

app.py

```
from flask import Flask,render_template, request, jsonify, session, redirect
#from flask_mysqldb import MySQL
from utils import category, find
from flask import cli
import ibm_db
import ibm_boto3
from ibm_botocore.client import Config, ClientError

COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="t81QlQrO4G_j-QNhbFWbI3QKQqsBgDCAH559eJ3v9MQK"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/22e0dc49422e4859bd2e7165396ce75a:cafe19de-eb19-4229-86d9-0204fcb0f8ff::"

# Create resource https://s3.ap.cloud-object-storage.appdomain.cloud
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

app = Flask(__name__)
app.secret_key = "vani"

#cloud
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tvb74810;PWD=EdGcG242OX2nZ522;","","")


#mysql
"""app.config['MYSQL_HOST'] = 'localhost'
```

```
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'flask'
mysql = MySQL(app)"""

@app.route('/')
def index():
  #files = get_bucket_contents('flaskapp123')
  return render_template('index.html') #, files = files

@app.route('/register')
def register():
  return render_template('register.html')

@app.route('/reg_res', methods = ['POST', 'GET'])
def reg_res():
   if request.method == 'GET':
      msg = "Please Login!!"
      return render_template("result.html", msg = msg)
   if request.method == 'POST':
    try:
      email = request.form['email']
      password = request.form['password']
      #cursor = mysql.connection.cursor()
      #cursor.execute('INSERT INTO user(email, password) VALUES(%s, %s)',(email,
password))
      #mysql.connection.commit()

      sql = "SELECT * FROM USER WHERE EMAIL=?"
      stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      user = ibm_db.fetch_assoc(stmt)



      if user:
        msg = "User already exist!!"
      else:
        jj
        sql = "INSERT INTO user(email, password) VALUES(?, ?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        msg = "User added successfully"

    except:
      msg = "error!! Can't add user"
      cursor.rollback()
    finally:
      return render_template("result.html", msg = msg)
      cursor.close()

@app.route("/login")
def login():
  return render_template('login.html')

@app.route("/log_res", methods = ['POST', 'GET'])
def log_res():
  if request.method == 'GET':
    msg = "Invalid Credentials!!"
    return render_template("result.html", msg = msg)
  if request.method == 'POST':
    try:
      email = request.form['email']
      password = request.form['password']
      #cursor = mysql.connection.cursor()
      #sql = "SELECT * FROM USER WHERE EMAIL=%s AND PASSWORD=%s",(email,
password)
      #cursor.execute(sql)
      #data = cursor.fetchone()
      #dbemail = data[1]
      #dbpassword = data[2]
      sql = "SELECT * FROM USER WHERE EMAIL=?"
```

```
         stmt = ibm_db.prepare(conn, sql)
         ibm_db.bind_param(stmt,1,email)
         ibm_db.execute(stmt)
         user = ibm_db.fetch_tuple(stmt)

         #if password == dbpassword:
         if user[2]==password:
            session['status']="ok"
            return render_template("home.html")
            #return redirect("http://127.0.0.1/home")
         else:
            msg = "Invalid Credentials!!"
            return render_template("result.html", msg = msg)
      except TypeError as e:
         msg = "Invalid Credentials!!"
         return render_template("result.html", msg = msg)

@app.route('/logout')
def logout():
    if 'status' in session:
        session.pop('status',None)
        msg = "logout successful!!"
        return render_template("result.html", msg = msg)
    else:
        msg = "User already logged out!!"
        return render_template("result.html", msg = msg)

@app.route('/home')
def getVariable():
   if 'status' in session:
      return render_template('home.html')
   else:
      msg = "Please Login!!"
      return render_template("result.html", msg = msg)

#countries
"""@app.route('/india')
```

```
def india():
    news_articles = country("in")
    return render_template("news.html", news_articles=news_articles)"""

#category
@app.route('/entertainment')
def entertainment():
    if 'status' in session:
        news_articles = category("in","entertainment")
        return render_template("news.html", news_articles=news_articles, cat="Entertainment")
    else:
        msg = "Please Login!!"
        return render_template("result.html", msg = msg)

@app.route('/sports')
def sports():
    if 'status' in session:
        news_articles = category("in","sports")
        return render_template("news.html", news_articles=news_articles, cat="Sports")
    else:
        msg = "Please Login!!"
        return render_template("result.html", msg = msg)

@app.route('/technology')
def technology():
    if 'status' in session:
        news_articles = category("in","technology")
        return render_template("news.html", news_articles=news_articles, cat="Technology")
    else:
        msg = "Please Login!!"
        return render_template("result.html", msg = msg)

@app.route('/science')
def science():
    if 'status' in session:
        news_articles = category("in","science")
        return render_template("news.html", news_articles=news_articles, cat="Science")
```

```
  else:
    msg = "Please Login!!"
    return render_template("result.html", msg = msg)

@app.route('/health')
def health():
  if 'status' in session:
    news_articles = category("in","health")
    return render_template("news.html", news_articles=news_articles, cat="Health")
  else:
    msg = "Please Login!!"
    return render_template("result.html", msg = msg)

@app.route('/business')
def business():
  if 'status' in session:
    news_articles = category("in","business")
    return render_template("news.html", news_articles=news_articles, cat="Business")
  else:
    msg = "Please Login!!"
    return render_template("result.html", msg = msg)

#search
@app.route('/search', methods = ['POST', 'GET'])
def search():
  if request.method == 'POST':
    if 'status' in session:
      sr = request.form['search']
      news_articles = find(sr)
      return render_template("news.html", news_articles=news_articles)
    else:
      msg = "Please Login!!"
      return render_template("result.html", msg = msg)

if __name__ == '__main__':
  app.run(port='90', debug=True)
```

utils.py

```
import requests
from decouple import config

NEWS_API_KEY = config('NEWS_API_KEY')

"""def country(country1):
    news_data = requests.get(f'https://newsapi.org/v2/top-headlines?country={country1}&apiKey={NEWS_API_KEY}').json()
    return news_data['articles']"""

def category(arg1,arg2):
    news_data = requests.get(f'https://newsapi.org/v2/top-headlines?country={arg1}&category={arg2}&apiKey={NEWS_API_KEY}').json()
    return news_data['articles']

def find(arg3):
    news_data = requests.get(f'https://newsapi.org/v2/everything?q={arg3}&apiKey={NEWS_API_KEY}').json()
    return news_data['articles']
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-9401-1659002289

PROJECT DEMO LINK:

https://drive.google.com/file/d/1tLbHiLVbpWEUIEsu7BPPokouG_W5VuDG/view?usp=share_link