

CLOUD BASED NEWS TRACKER APPLICATION

PROJECT REPORT

Submitted by

SANTOSH B (19EUCS122)

KIRAN S (20EUCS503)

MAHARANTH H (19EUCS078)

MITHILESH P C (19EUCS085)

in partial fulfillment of the requirements for the award of

the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND

TECHNOLOGY COIMBATORE

(An Autonomous Institution)



ANNA UNIVERSITY:

CHENNAI MAY 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University, Chennai)

ACCREDITED BY NAAC WITH “A” GRADE

BONAFIDE CERTIFICATE

Certified that this project report titled **“Cloud based News tracker Application”** is the bonafide work of Santosh B (19EUCS122), Kiran S (20EUCS503), Maharanth H (19EUCS078), Mithilesh P C (19EUCS111) who carried out the project work under my supervision.

SIGNATURE

Dr.K. SASI KALA RANI, M.E., Ph.D.,

M.E

M.EHEAD OF THE DEPARTMENT

SIGNATURE

Majidha Fathima K M

SUPERVISOR

Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Kuniamuthur,
Coimbatore

This project report is submitted for the Autonomous Project Viva-Voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our sincere thanks to the management and **Dr.J.JANET, M.E.,Ph.D.,** Principal, Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this project work.

We are highly indebt to **Dr.K. SASIKALA RANI, M.E.,Ph.D.,** Head of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We are thankful to **Dr. Vijeya Kaveri V, M.E.,Ph.D.,** Project Co-ordinator, Department of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We express our deep sense of gratitude to our guide, **Majidha Fathima K M**

M.E, Professor in the department of Computer science and Engineering for her valuable advice, guidance and support during the course of our project work.

By this, we express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have all helped in collecting the resources and materials needed for this project and for their support during the study and implementation this project.

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 Deninition
- 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 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
7. CODING & SOLUTIONING
 - 7.1 Feature 1
 - 7.2 Feature 2
8. TESTING
 - 8.1 'Test Cases
 - 8.2 User Acceptance Testing
9. RESULTS
 - 9.1 Performance Metrics
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX
 - Source Code
 - GitHub & Project Demo Link

1. INTRODUCTION

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

1.1 PROJECT OVERVIEW

To develop an news application which provides the quality news based on the user interest. Here there are many sections and those section's news are tracked from the news sources and the user could select their required section and they could read the news that are given as a snippet and that will be enough for a user to gain an knowledge about that particular news. It is a cloud based application that it is mainly used for the users who hasn't have their enough time for their daily updation. It gives the quality news which will be provided by the best news sources and also the news will be displayed according to the users interests ,so that they could read easily.

1.2 PURPOSE

It is a cloud based application that it is mainly used for the users who hasn't have their enough time for their daily updation. It gives the quality news which will be provided by the best news sources and also the news will be displayed according to the users interests ,so that they could read easily.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

THE GENDER GAP TRACKER: Using Language Processing to Gender Bias in media. This study provided an opportunity to determine the type of information that can be provided by analysis of Students' on-line interactions. Usage patterns of the WIER website and resources in terms of frequency of access and time spent at the site provided a rich picture of student learning behaviour.

2.2 REFERENCES

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

Authors : Yajie Qi, Li Zhou, Huayou Si, Jian Wan, Ting Jin.

About the paper :

Once an event occurs, usually there are a large number of online news to be released. How to quickly and accurately detect the hot events from the huge amount of online news is the focus and hotspot.

2. Exploring Mobile News Reading Interactions for News App Personalisation

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

About the paper :

As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces.

3. Android News App

Authors: Brijesh Joshi, Nehal Patel.

About the paper :

As world's technology is rapidly growing, we have fast connection and network to instantly connect to other person. Day to day use in mobile, tablets and laptop is increasing, most of the people already have this facilities.

4. Research on Topic Detection and Tracking for Online News Text

Authors: Guixian Xu, Yueting Meng, Zhan Chen, Xiaoyu Qiu, Changzhi Wang, Haishen Yao.

About the paper :

With the rapid development of the Internet, the amount of data has grown exponentially. On the one hand, the accumulation of big data provides the basic support for artificial intelligence. On the other hand, in the face of such huge data information, how to extract the knowledge of interest from it has become a matter of general concern.

2.3 PROBLEM STATEMENT DEFINITION

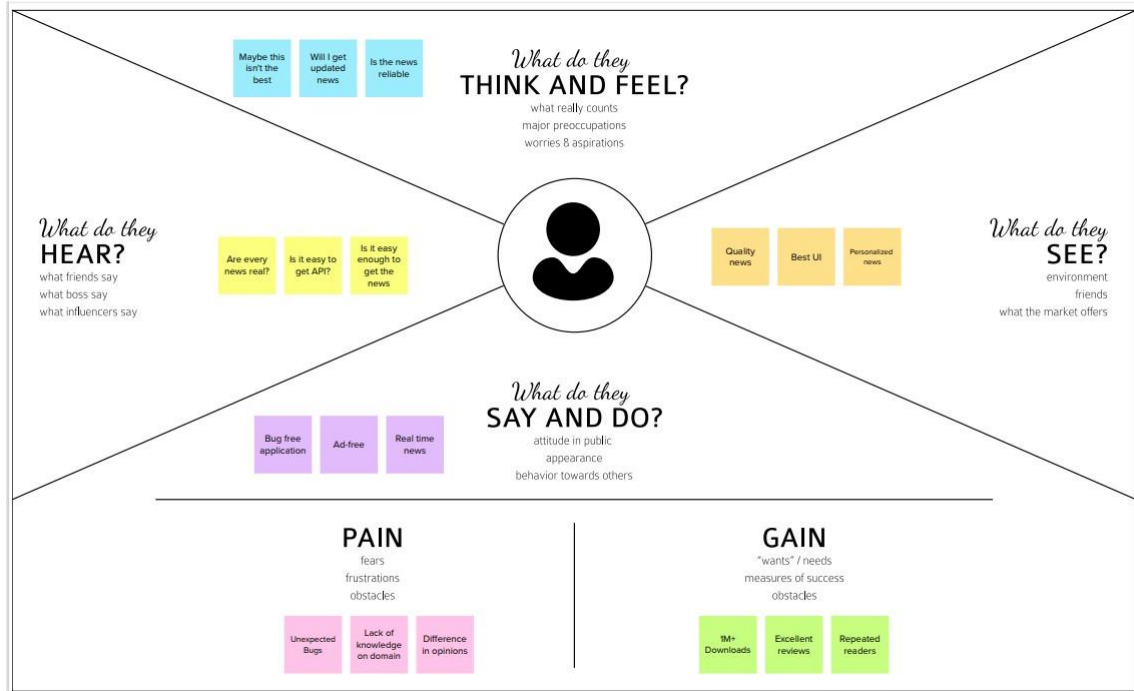
It is a cloud based application that it is mainly used for the users who hasn't have their enough time their daily updation. It gives the quality news which will be provided by the best news sources and also a news will be displayed according to the users interests ,so that they could read easily.

To develop an news application which provides the quality news based on the user interest. Here there are many sections and those section's news are tracked from the news sources and the user could select their required section and they could read the news that are given as a snippet and that will be enough for a user to gain an knowledge about that particular news.

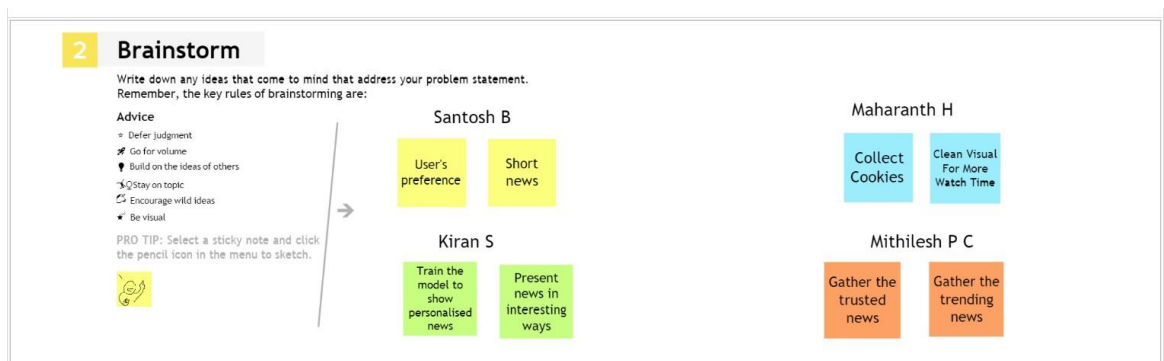
TITLE	MECHANISM USED	ADVANTAGES	DISADVANTAGES
Paper 1: Design and development of cloud based news application.	Facebook's Parse.com for backend.	It will provide the first hand information to the users.	The UI is little much difficult.
Paper 2: Exploring mobile news reading interactions for news app personalisation	5clustering techniques <ul style="list-style-type: none">• Frequency• Daily reading• Browsing strategy.• Reading style• Location.	To recommend right news to right user.	The design of adaptive interface is missing.
Paper 3: Tracking of data leaving the cloud	CloudDT-Data accountability framework is used to track the data.	Detective work in finding the data that leaving the cloud.	Loss of data due to uncontrolled environment.

3. IDEATION AND PROPOSED SYSTEM

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



3

Group ideas

The facilitator should group all the ideas from the brainstorming process (step 2). After that, you should add your opinions by adding arrows to point ideas into other groups and sticky notes and icons to share your thoughts.

PRO TIP: This is a great place to use color coding. You can change the color of multiple sticky notes at once.

Data collection

With the use of Cookies we collect data of users

Ask users what topic interest them

Data classification

user's interest is stored in database

classified based on the nature of the news.

Data features

Understand which news a user views frequently

Flash news is updated frequently

Evaluation of data

News will be updated time to time

Fake news will be eliminated

4

Prioritize

The facilitator should copy and paste the groups from step 3 into this area and setup the vote details.

Let's vote! Add a vote on sticky notes you think are a high priority. You can also add your vote to an entire group.

Collecting a trending news

Specific News Source.

Deleting old contents

Type of News and Languages

Best UI

Content of the news should match user's interest

Quality of news should be very high

Update user interests



3.3 PROPOSED SOLUTION

PROBLEM STATEMENT (PROBLEM TO BE SOLVED)

It is a cloud based application that it is mainly used for the users who hasn't have their enough time their daily updation. It gives the quality news which will be provided by the best news sources and also a news will be displayed according to the users interests ,so that they could read easily.

IDEA / SOLUTION DESCRIPTION

We are using html - bootstrap ,python - flask. This is for the continuous display of news. As Python has a lot of libraries, it is easier and faster. Provides an analyzed insight into day to day news. It gives unbiased news in an easy and orderly manner. News today is ridiculed with adverts and sponsorships which fail to provide unparalleled news. Our solution prevents adverts therefore providing users unbiased news. Our solution aims to provide new news up to date in real-time with personal suggestions.

NOVELTY / UNIQUENESS

Usually people use javascript ,php as a backend for web development. But we are using flask for that purpose. It predicts output with high accuracy. App-based news tracker dashboard systems which can analyze real-time images of an incident and analyze it for news content which can be very handy and useful.

SOCIAL IMPACT / CUSTOMER SATISFACTION

Less time and more knowledge consumption.

BUSINESS MODEL (REVENUE MODEL)

A revenue model means understanding how a startup can make money. Our major revenue sources consist of sales, government funds, and public donations.

SCALABILITY OF THE SOLUTION

Design, news and flash news are flexible. IBM Cloud, for instance, is one of the cloud-based AI scalability options. Run and manage AI models, as well as optimise decisions at scale across any cloud, with the aid of IBM Cloud Build. The benefit of using the cloud to scale solutions is that we can install our AI programme there. the specific cloud environment that best supports our business needs. We can take advantage of built-in security capabilities and AI model monitoring. we can drive better business outcomes by optimizing our decisions and also make our solution scalable using cloud.

3.4 PROBLEM SOLUTION FIT

Cloud Computing is one of the fast growing and it offers flexible resources, and economies of scale. In this article we are going make an application which tracks the news. It is an News Tracker Application where it gives crisp news from the reputed news sources and it's just gives the simple sentence which saves the readers time and also it is high in knowledge.

PURPOSE:

Solve complex problems in a way that fits the state of your customers. Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior. Sharpen your communication and marketing strategy with the right triggers and messaging. Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems. Understand the existing situation in order to improve it for your target group.

Project Title: NEWS Tracker Application		Project Design Phase-I - Solution Fit Template		Team ID: PNT2022TMID02787	
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Working individuals and passionate NEWS readers	6. CUSTOMER CONSTRAINTS CC For ad free interface every news provider demand subscription from users	5. AVAILABLE SOLUTIONS AS Used Adblockers to avoid ad, but it's not secure for the user	Explore AS, differentiate	
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Clumsy UI, Not Personalized, Vague news	9. PROBLEM ROOT CAUSE RC Now a days news has become a money making tool for the news agency so they force customer to pay for subscription or use ad loaded free version	7. BEHAVIOUR BE Pay for subscription or adjust with the ads shown		
Identify strong TR & EM	3. TRIGGERS TR Seeing others using NEWS app with ad free interface motivate users to download the same	10. YOUR SOLUTION SL Provide free access to ad-free UI to read news, by using free APIs available to Provide personalized news to the users	8. CHANNELS of BEHAVIOR CH <u>ONLINE</u> Uses memes and posts in social media to get updated about happenings <u>OFFLINE</u> Talk with peoples to know the current events and happenings	Identify strong TR & EM	

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

User Registration

- Registration through Form
- Registration through Gmail

User Confirmation

- Confirmation via Email
- Confirmation via OTP

Searching

- Search the news based on the user interest.
- Trending news will be shown.
- User can choose the genre.

Real-time News

- User could see the real news from trusted sources.
- Auto update news because it fetch news from NEWS-API.

Location Based

- User could see their country news.
- They could also see other country news.

4.2 NON-FUNCTIONAL REQUIREMENTS

PERFORMANCE

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design an application, which will fit into the required environment. The load for the user interface screens shall take no longer than 2 seconds. The login information shall be verified within 5 seconds. Queries shall return results within 5 seconds.

Usability

- The news is updated almost every second, it does not affect the user who reads his/her favourite news.

Security

- Authentication and Password Management.

Reliability

- Avoid fake news.
- Instant news.
- Location based news.

Performance

- Smooth UI makes the user to stay in app.
- Showing the current trending news first.

Availability

- Minimum idle time is 24/7.
- Recent news will based on user's previous news visit.

Scalability

- Get more user's attention by sharing the app in social media

SAFETY AND SECURITY REQUIREMENTS

- User Identification:

The system requires the user to identify himself/herself User

- Login ID:

Any user who uses the system shall have a Login.

- Modification:

Any modification (insert, delete (or) update) for the Database shall be synchronized and done only by the admin in the ward.

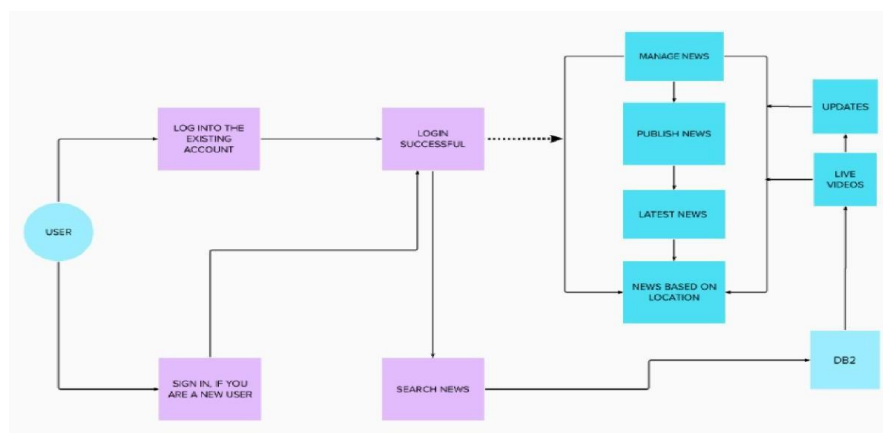
- Admin Rights:

Admin shall be able to view and modify the information.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

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



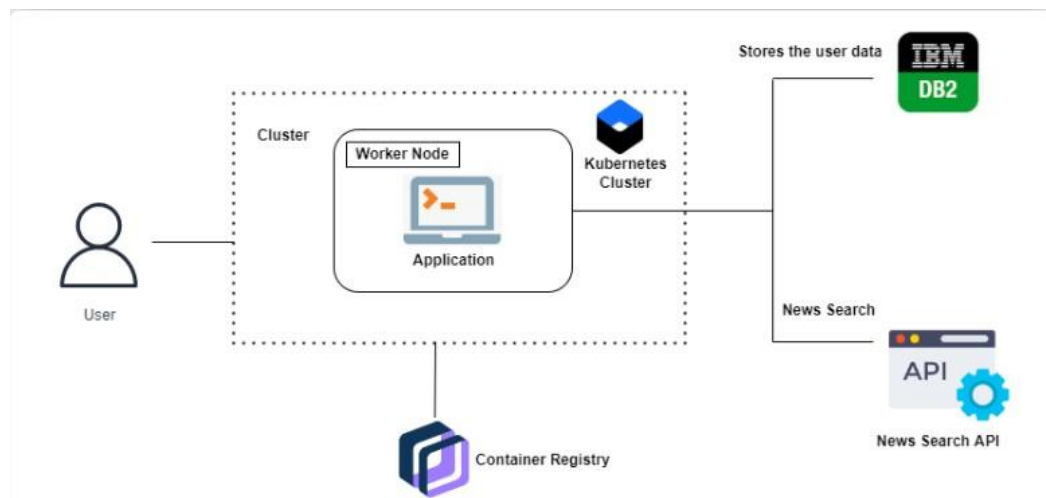
5.2 SOLUTION AND TECHNICAL ARCHITECTURE

PROJECT DESCRIPTION:

It is a cloud based application that it is mainly used for the users who hasn't have their enough time their daily updation. It gives the quality news which will be provided by the best news sources and also a news will be displayed according to the users interests ,so that they could read easily.

To develop an news application which provides the quality news based on the user interest. Here there are many sections and those section's news are tracked from the news sources and the user could select their required section and they could read the news that are given as a snippet and that will be enough for a user to gain an knowledge about that particular news.

TECHNICAL ARCHITECTURE



SOLUTION

- The user interacts with the application.
- Registers by giving the details.
- Integrate the application with news APIs and store the data in the database.
- The database will have all the details and the user can search the news by using a search bar.

PROCEDURE

Registration :

As a user, I can register for the application by entering my email, password, and confirming my password.

Login :

As a user, I can log into the application by entering email & password

Dashboard :

The news portal fetches for the most recent news and shows it as “Breaking News”.

Search Bar :

User search data based on their personal interests.

Administrator Server :

Provides the exact news from the database.

APPROACH:

KUBERNETES CLUSTERS - Kubernetes clusters allow containers to run across multiple machines and cloud based application.

IBM DB2- Used for Backup & recovery. Comprehensive data resilience for physical and virtual servers.Cloud hosting. Dedicated, virtual private, and bare metal server options

CONTAINER REGISTRY - Container Registry is a single place for your team to manage Docker images, perform vulnerability analysis, and decide who can access what with fine-grained access control

USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	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
	Login	USN-2	As a user, I can log into the application by entering email & password	I can start reading news.	High	Sprint-1
	Dashboard	USN-3	The news portal fetches for the most recent news and shows it as "Breaking News".	I can open and view the "Breaking News"	Medium	Sprint-2
	Search Bar	USN-4	User search data based on their personal interests	I can view the related content for the search data	Hign	Sprint-2
		USN-5	The news is showed with a sample snapshot and mini description and user can open and read the article.	I can view the news details and can open to see whole article	Medium	Sprint-3
Administrator	Server	USN-6	Provides the exact news from the database	I can see the latest news getting refreshed	Medium	Sprint-4
		USN-7	Provides Personalized News based on the intrest	I can see the news based on my interest	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

.PRODUCT BACKLOG, SPRINT SCHEDULE, AND ESTIMATION

Sprint	Functional Requirements(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	15	High	Santosh B, Kiran S, Mithilesh P C, Maharanth H
Sprint 2	Login	USN-2	As a user, I can log into the application by entering email & password	10	High	Santosh B, Kiran S, Mithilesh P C, Maharanth H
Sprint 3	Database	USN-3	Connection to the database for maintaining the user details.	10	High	Santosh B, Kiran S, Mithilesh P C, Maharanth H
Sprint 4	Notifications	USN-4	As a user, I will receive notifications to my email so that I'll be updated on the news	10	Medium	Santosh B, Kiran S, Mithilesh P C, Maharanth H

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	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

VELOCITY:

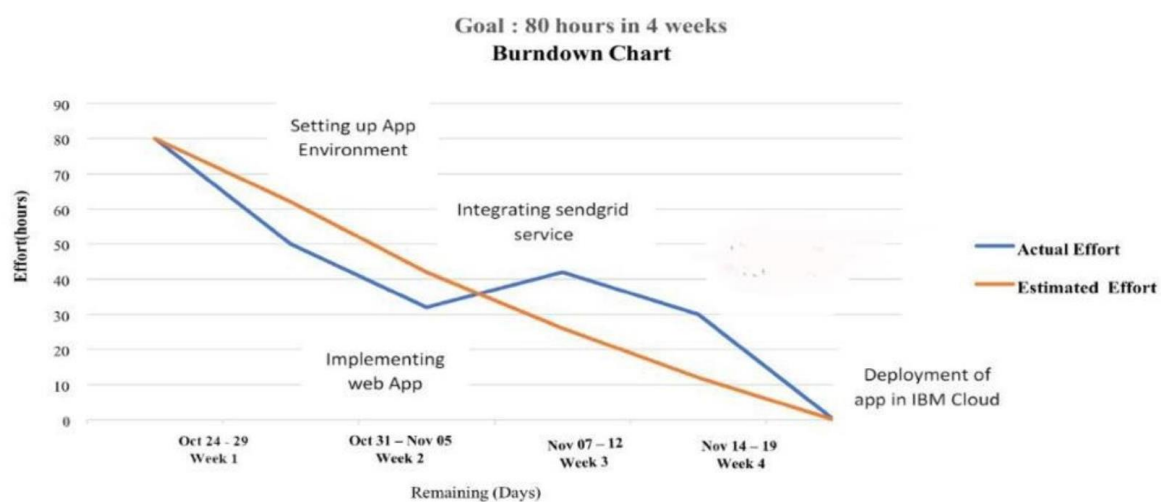
Imagine we have a 10-day sprint duration, and the velocity of the team 20(points per sprint).Let's calculate the team's average velocity (AV) per iteration unit.

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

6.2 SPRINT DELIVERY SCHEDULE

Milestone List	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct	29-Oct	30-Oct	31-Oct	1-Nov	2-Nov	3-Nov	4-Nov	5-Nov	6-Nov	7-Nov	8-Nov	9-Nov	10-Nov	11-Nov	12-Nov	13-Nov	14-Nov	15-Nov	16-Nov	17-Nov	18-Nov	19-Nov
Sprint 1 - Registration and Sign in																											
Design Sign Up & Sign in Page	1 Day																										
Email Auth		2 Days																									
DB2 Database Design			1 Day																								
Email and Password Sign in				1 Day																							
Email Confirmation on user account creation					1 Day																						
Sprint 2 - API Fetching and Backend Endpoints																											
Fetch data from Rapid API					2 Days																						
Flask-REST API coding						2 Days																					
Create timed function for fetch from API using Threading									1 Day																		
Test backend										2 Day																	
Sprint 3 - UI and UX design and Connecting frontend and backend																											
Design main Welcome Page													3 Days														
News Card Design														2 Days													
Explore Designs and Saved Design																	1 Day										
Bookmarks design																		1 Day									
Connecting Frontend and backend																			1 Day								
UI responsiveness																				2 Days							
Sprint 4 - Deployment, Testing and Integrations																											
Deploying the App on cloud																					3 Days						
Implementing Loggers																								2 Days			
Final Testing																										2 Days	

6.3 REPORT FROM JIRA



7. CODING AND SOLUTIONING

7.1 FEATURE 1

Base.html

```
<!DOCTYPE
  YPE
  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">
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/u
md/popper.min.js"
    integrity="sha384-
7+zCNj/IqJ95wo16oMtfSKbZ9ccEh31eOz1HGYDuCQ6wgnyJ
NSYdrPa03rtR1zdB"
    crossorigin="anonymous"></script>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/boot
strap.min.css"
    integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlf
IDPvg6uqKI2xXr2" crossorigin="anonymous">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstr
ap.min.js"
    integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEF
EIsxhlmWl5/YESvpZ13"
    crossorigin="anonymous"></script>
    <link rel="stylesheet" href="../static/app.css">

    <!-- Added absolute path to fix rendering problems when in
nested route. check to ensure works on deployment -->
    <script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

```

<title>NewsTracker</title>
<link href="https://fonts.googleapis.com/css?family=Abel"
rel="stylesheet">
</head>

<body>
  <nav class="navbar border-bottom border-secondary fixed-top
navbar-expand navbar-dark bg-dark">
    <a href="/" class="navbar-brand">NewsTracker</a>
    <div class="collapse navbar-collapse" id="navbarNav">
      <div class="navbar-nav">
        {% if g.user %}
          <a class="nav-item nav-link "
href="/headlines">Headlines <span class="sr-
only">(current)</span></a>
          <a class="nav-item nav-link" href="/user/saved">My
Stories</a>
          <a class="nav-item nav-link" href="/search">Detailed
Search</a>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#"
id="navbarDarkDropdownMenuLink" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
My Queries
</a>
            <ul class="dropdown-menu dropdown-menu-dark" aria-
labelledby="navbarDarkDropdownMenuLink" id="dropdown">
              {% if g.user.queries%}
              {%for query in g.user.queries%}
                <li class="dropdown-item" style="
padding-right: 10px"><a class="query-link"
href="/search/{{query.id}}">{{query.name}}
</a>
                <span data-query="{{query.id}}"
class="close">&times;</span>
                <!-- <button type="button" class="btn btn-outline-
success btn-sm">Make Default</button> -->
              </li>
              {%endfor%}
              {% else %}
                <span class="dropdown-item">You currently have no
saved queries.</span>

```

```

        {% endif % }
    </ul>
</li>
<a class="nav-item nav-link" href="/logout">Logout</a>
{% else % }
<a class="nav-item nav-link"
href="/headlines">Headlines</a>
<a class="nav-item nav-link" href="/login">Login</a>
<a class="nav-item nav-link"
href="/register">Register</a>
{% endif % }
</div>
</div>
<!-- <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav">
    <span class="navbar-toggler-icon"></span>
</button> -->
<div>
    <form action="/search/simple" method="GET"
class="form-inline" id="search">
        <input name="search" class="form-control mr-sm-2"
type="search" placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success badge-pill my-2
my-sm-0 badge-pill" type="submit">Search</button>
    </form>
</div>
</nav>
<div id="indentation">
</div>
{% with messages =
get_flashed_messages(with_categories=true) % }
{% if messages % }
<section class="messages">
    {% for category, msg in
get_flashed_messages(with_categories=true)% }
        <span class="{{ category }}">{{ msg }}</span>
    {% endfor % }
</section>
{% endif % }
{% endwith % }

{% block content % }
<!-- content goes here -->

```

```

    {% endblock %}
    <script src="/static/newstracker.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/po
pper.min.js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3U
ksdQRVvoxMfooAo"
    crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-
J6qa4849bIE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFA
V0wwj1yYfoRStJoZ+n" crossorigin="anonymous">
    </script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/boots
trap.min.js"
    integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uo
d8GCExl3Og8ifwB6"
    crossorigin="anonymous"></script>
</body>

</html>

```

Login.ht
ml

```

{%
extends
'base.htm
l' %}

```

```

{% block content %}
<div class="container">
    <h1 class="form display-3 text-center">Login</h1>
    <div class="text-center text-light">
        <p class="display-5 alert alert-success" id="login-q">Don't
have an account? <a href="/register"
class="alert-link">Register here</a></p>
    <div class="row justify-content-center">
        <div class="col-md-8">
            <form method="POST">
                {{ form.hidden_tag() }}

```

```

        {% for field in form if field.widget.input_type !=
'hidden' %}
        <p>
        <div class="form-group">
            {{ field.label }}
            {{ field(class_="form-control") }}
            {% for err in field.errors %}
            <small class="form-text text-danger">
                {{ err }}
            </small>
            {% endfor %}
        </div>
        </p>
        {% endfor %}
        <button class="submit-b btn btn-success btn-large"
type="Submit">Login</button>
    </form>
</div>
</div>
</div>
</div>
{% endblock %}

```

7.2 FEATURE 2

App.py

```

from dotenv
import
load_dotenv

load_dotenv()
from flask import Flask
app = Flask(__name__)
if app.config["ENV"] == "production":
    app.config.from_object('config.ProductionConfig')
elif app.config["ENV"] == "development":
    app.config.from_object('config.DevelopmentConfig')
else:
    app.config.from_object('config.TestingConfig')

```



```

#server-side session
CURR_USER_KEY = "curr_user"
from flask_session import Session
server_session = Session(app)

#db set-up
from models import connect_db, db
connect_db(app)
db.create_all()

#view imports
from views import sa_views, site_views, api

```

Config.py

```

import
os

import redis

class Config(object):
    DEBUG = False
    TESTING = False
    SQLALCHEMY_TRACK_MODIFICATIONS = False
    SQLALCHEMY_ECHO = True
    API_KEY = os.getenv("API_KEY")
    SECRET_KEY = os.getenv("SECRET_KEY")
    SESSION_TYPE = 'redis'
    SESSION_USE_SIGNER = True
    SESSION_PERMANENT = False

class DevelopmentConfig(Config):
    DEBUG = True
    SQLALCHEMY_DATABASE_URI = 'postgresql:///news-tracker7'
    SESSION_REDIS = redis.from_url('redis://localhost:6379')

class ProductionConfig(Config):
    SQLALCHEMY_DATABASE_URI =
os.getenv("DATABASE_URL")
    SESSION_REDIS = redis.from_url(os.getenv('REDIS_URL',
'redis://localhost:6379')) #2nd argument for local db is necessary for

```

some reason because the production version keeps getting called in development.

```
class TestingConfig(Config):
    TESTING = True
    SQLALCHEMY_DATABASE_URI =
'postgresql:///newstracker-test'
    SQLALCHEMY_ECHO = False
    DEBUG_TB_HOSTS= ['dont-show-debug-toolbar']
    WTF_CSRF_ENABLED = False
```

Forms.py

```
from
flask_wtf
import
FlaskForm

from wtforms import StringField, PasswordField, IntegerField,
SelectField, DateField, BooleanField, DateTimeField
from wtforms.validators import InputRequired, Optional,
NumberRange
import datetime as dt
```

```
class RegisterForm(FlaskForm):
    """form to add a new user. Username, password, and email
    required"""
    username = StringField("Username",
validators=[InputRequired()])
    password = PasswordField("Password",
validators=[InputRequired()])
    email = StringField("Email")
    first_name = StringField("First Name")
    last_name = StringField("Last Name")
```

```
class LoginForm(FlaskForm):
    """form to login user with username and password"""
    username = StringField("Username", validators=[
        InputRequired(message="Please enter a
username")])
    password = PasswordField("Password", validators=[
```

```
InputRequired(message="Please enter a
password"))])
```

```
class SearchForm(FlaskForm):
```

```
    keyword = StringField("Enter a search term:",
validators=[Optional()])
```

```
    source = SelectField("Choose source:", choices=[
        ("", 'All'), ('abc-news', 'ABC News'), ('al-jazeera-english',
        'Al Jazeera'), ('associated-press',
        'Associated Press'), ('axios', 'Axios'),
        ('bbc-news', 'BBC News'), ('bloomberg', 'Bloomberg'),
        ('cbc-news',
        'CBC News'), ('cbs-
        news', 'CBS News'), ('cnn', 'CNN'), ('fox-news', 'Fox News'),
        ('google-news', 'Google News'), ('independent',
        'Independent'), ('msnbc',
        'MSNBC'), ('nbc-
        news', 'NBC News'), ('politico', 'Politico'),
        ('reuters', 'Reuters'), ('the-hill', 'The Hill'), ('the-huffington-
        post',
        'The Huffington Post'),
        ('the-wall-street-journal', 'The Wall Street Journal'),
        ('the-washington-post', "The Washington Post"), ('time',
        'Time'), ('usa-today', 'USA Today'), ('vice-news', 'Vice News')],
        validators=[Optional()])
```

```
    quantity = IntegerField("Enter # of many articles you want
returned (max 15):",
        default=10, validators=[
        NumberRange(min=1, max=15, # this is represented by the
        "pageSize" parameter in the API where default and max are
        both set to 100
        message="Please enter a number between 1 and 10"))])
```

```
    date_from = DateField("Select a starting date:",
validators=[Optional()])
    date_to = DateField("Up until which date?", # API default if
will be newest to oldest with no limit
        default=dt.datetime.today, validators=[Optional()])
```

```
language = SelectField("Choose which language you want to  
see results from:", choices=[ # API default is all languages
```

```
    ('en', 'English'), ('es', 'Spanish'), ('fr', 'French'),  
    ('ar', 'Arabic'), ('he', 'Hebrew'), ('he', 'Hebrew'), ('it',  
'Italian'),  
    ('nl', 'Dutch'), ('no', 'Norwegian'), ('pt', 'Portuguese'), ('ru',  
'Russian'),  
    ('zh', 'Chinese'), ('se', 'Swedish')],  
    validators=[Optional()]) # can't find 'ud' language code as  
specified by api
```

```
sort_by = SelectField("Search by:", choices=[('relevancy',  
'Relevant'), ('popularity', 'Popular'), ('publishedAt', 'Recent'), #  
API default: publishedAt
```

```
    ('polarity', 'Polarity (results  
ordered from positive to negative)'),  
    ('subjectivity', 'Objectivity  
(results ordered from objective to subjective)')],  
    validators=[Optional()])
```

```
default = BooleanField(  
    "Make this your default search settings for your home page  
feed?")
```

```
saved_query = BooleanField(  
    "Would you like to add this to your saved search  
queries?")
```

```
name = StringField("Give your saved search query a name (if  
applicable)", validators=[Optional()])
```

Newstracker.js

```
const body = document.querySelector("body");
```

```
const homeAdd = function (arg) {  
    arg.classList.add("home-body");  
}
```

//axios requests that reach our server to run SA on said story and show result inside button

```
const saButtons = document.querySelectorAll(".sa-button");
for (let button of saButtons) {
  button.addEventListener("click", async function (evt) {
    evt.preventDefault();
    const text = button.firstChild;
    const input = text.nextSibling;
    const value = input.value;
    const storyID = button.getAttribute('data-story')
    if (value === "Get Polarity") {
      try {
        const req = await axios.post(`/story/${storyID}/polarity`);
        const resp = req.data.response;
        input.value = resp;
      } catch (error) {
        console.error(error.response.data)
      }
    }
    else if (value === "Get Subjectivity") {
      const req = await axios.post(`/story/${storyID}/subjectivity`);
      const resp = req.data.response;
      input.value = resp;
    }
  })
}
```

```
const deleteQueryButtons = document.querySelectorAll(".close")
for (let button of deleteQueryButtons) {
  button.addEventListener("click", async function (evt) {
    console.log("CLICKEDDDDD")
    const queryID = button.getAttribute('data-query')
    console.log(queryID)
    const req = await axios.post(`/user/${queryID}/delete`);
    const resp = req.data.response;
    button.value = resp;
  })
}
```

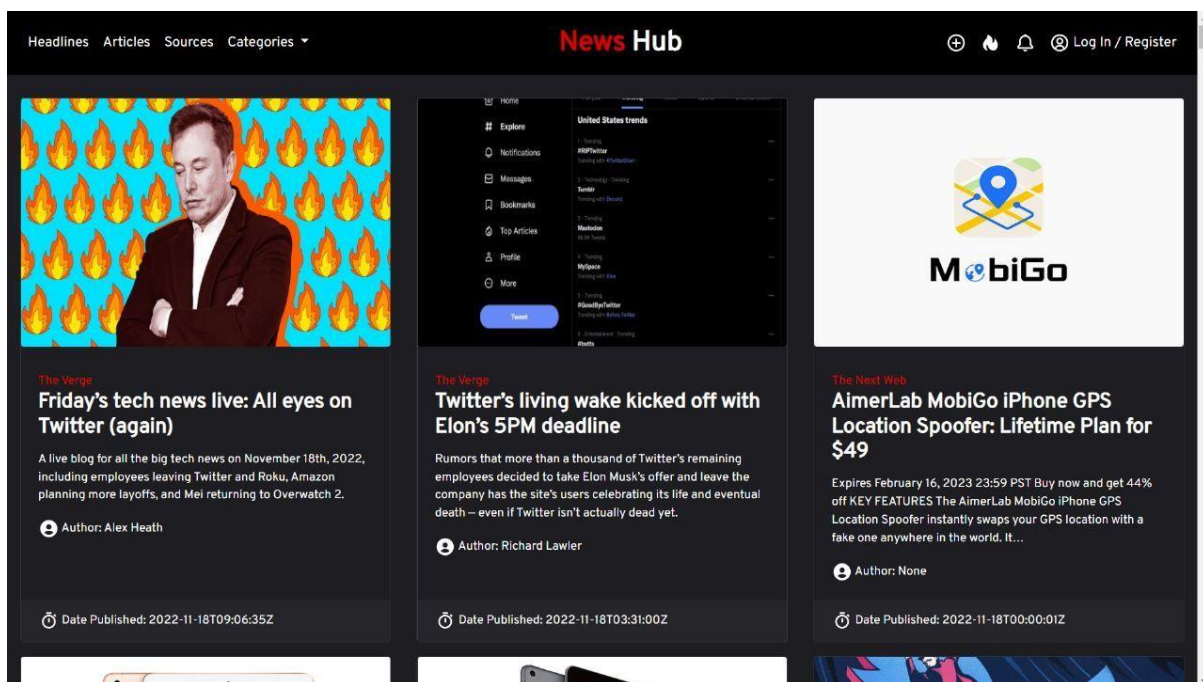
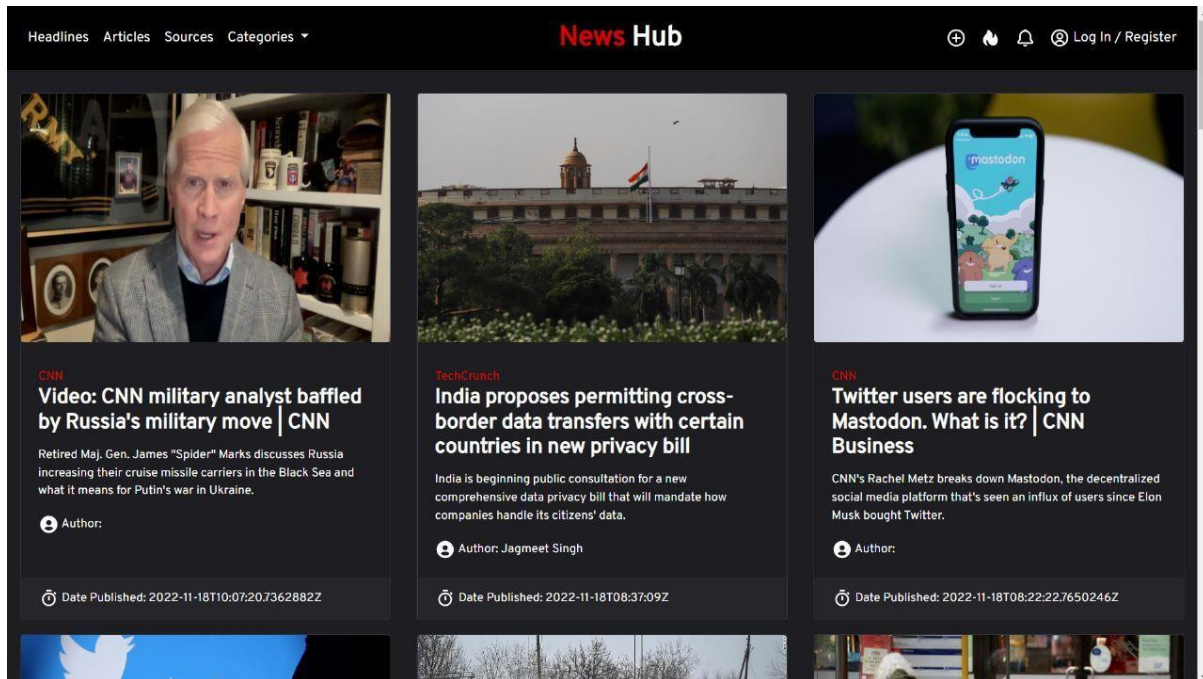
//Replace broken images with default image

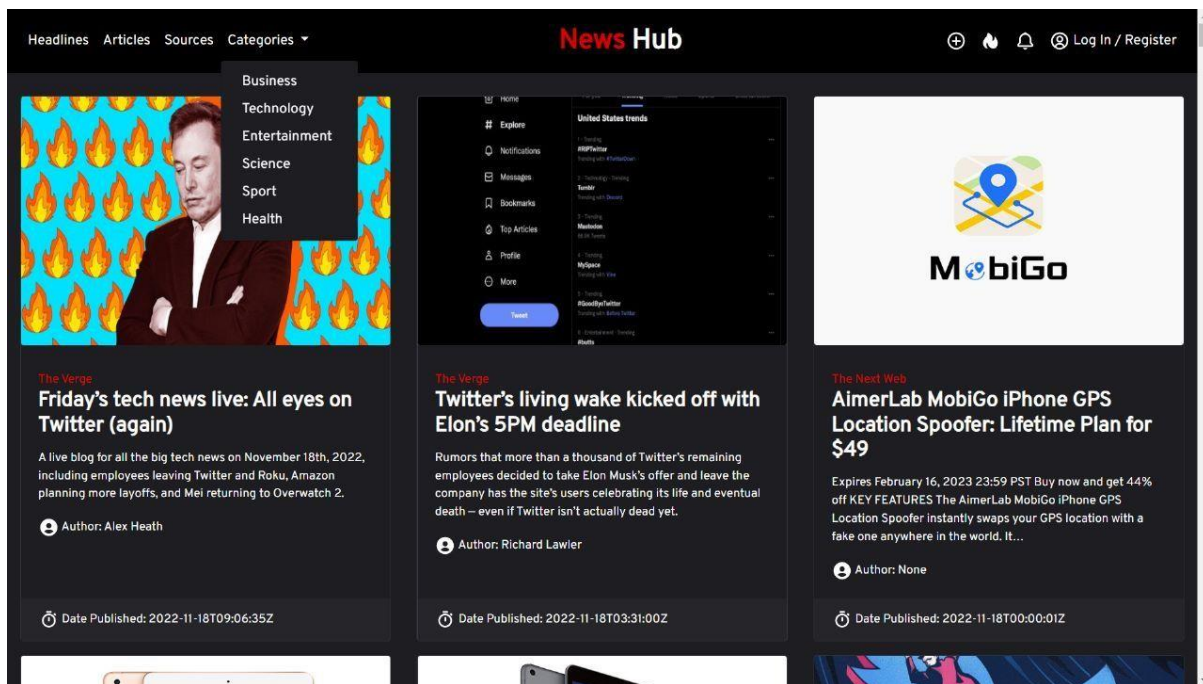
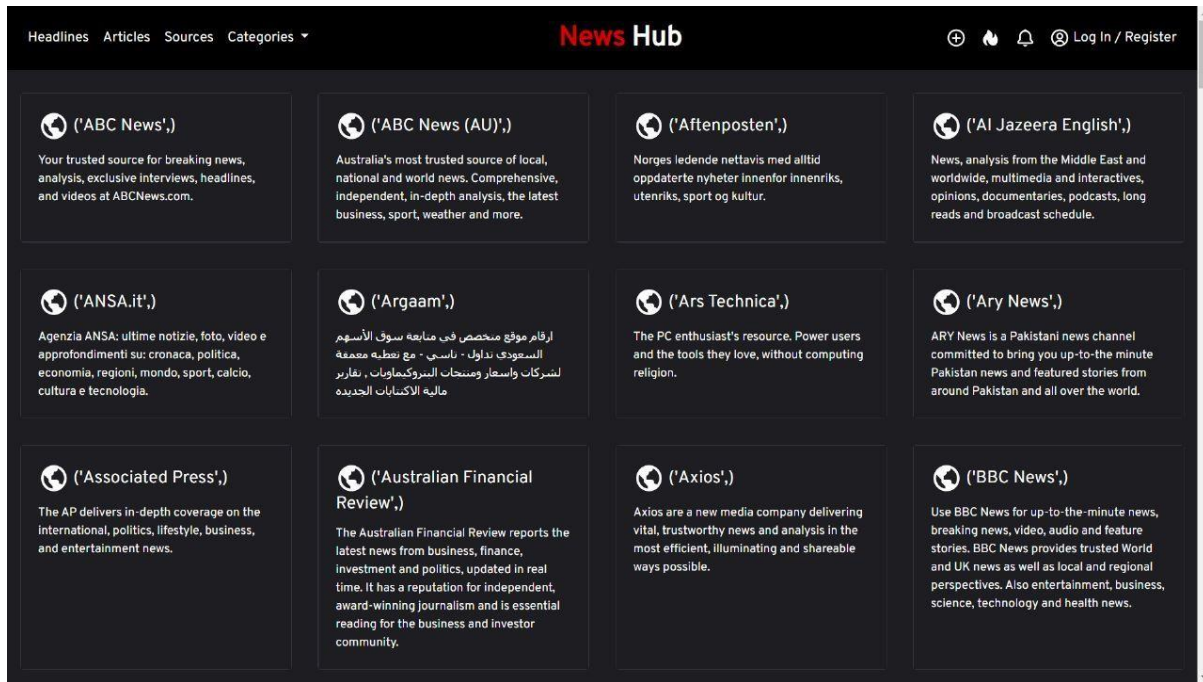
```
const default_avatar = 'https://secure.gravatar.com/avatar?d=wavatar';

window.addEventListener("load", event => {
  let images = document.querySelectorAll('img');
  for (let image of images) {
    let isLoaded = image.complete && image.naturalHeight !== 0;
    if (!isLoaded) {
      image.src = default_avatar;
    }
  }
});
```

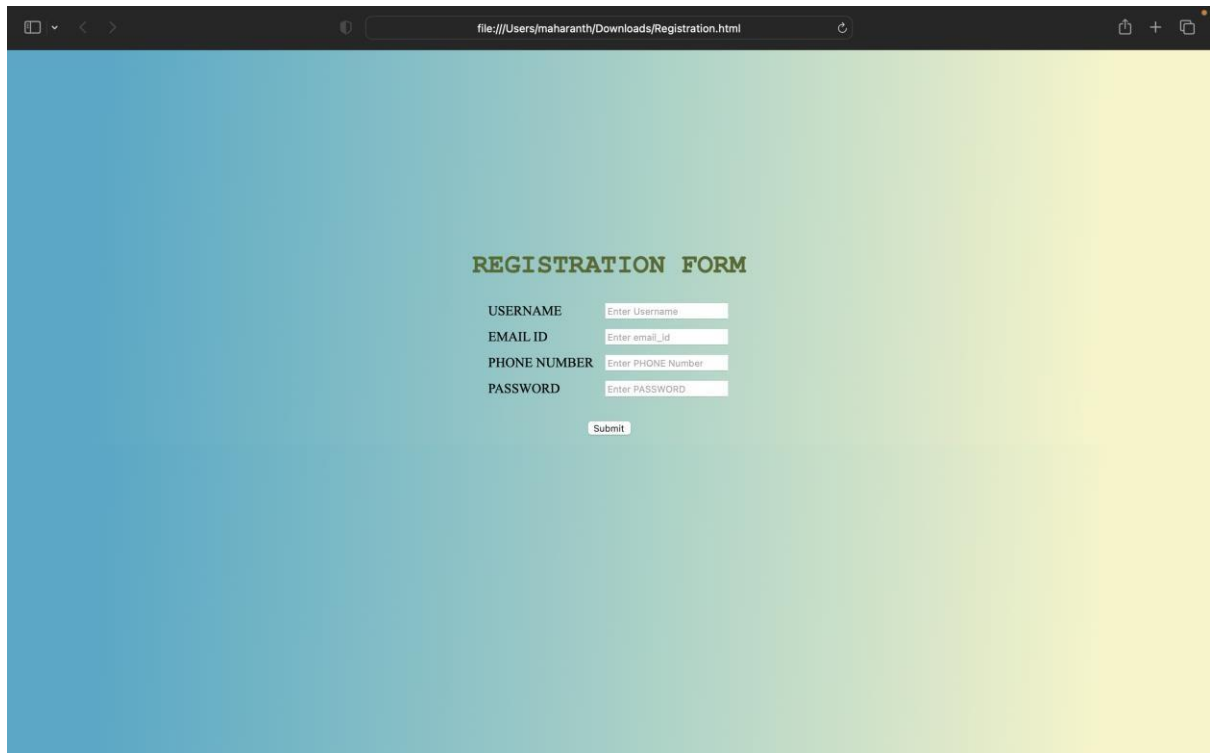
8. TESTING

8.1 TEST CASES

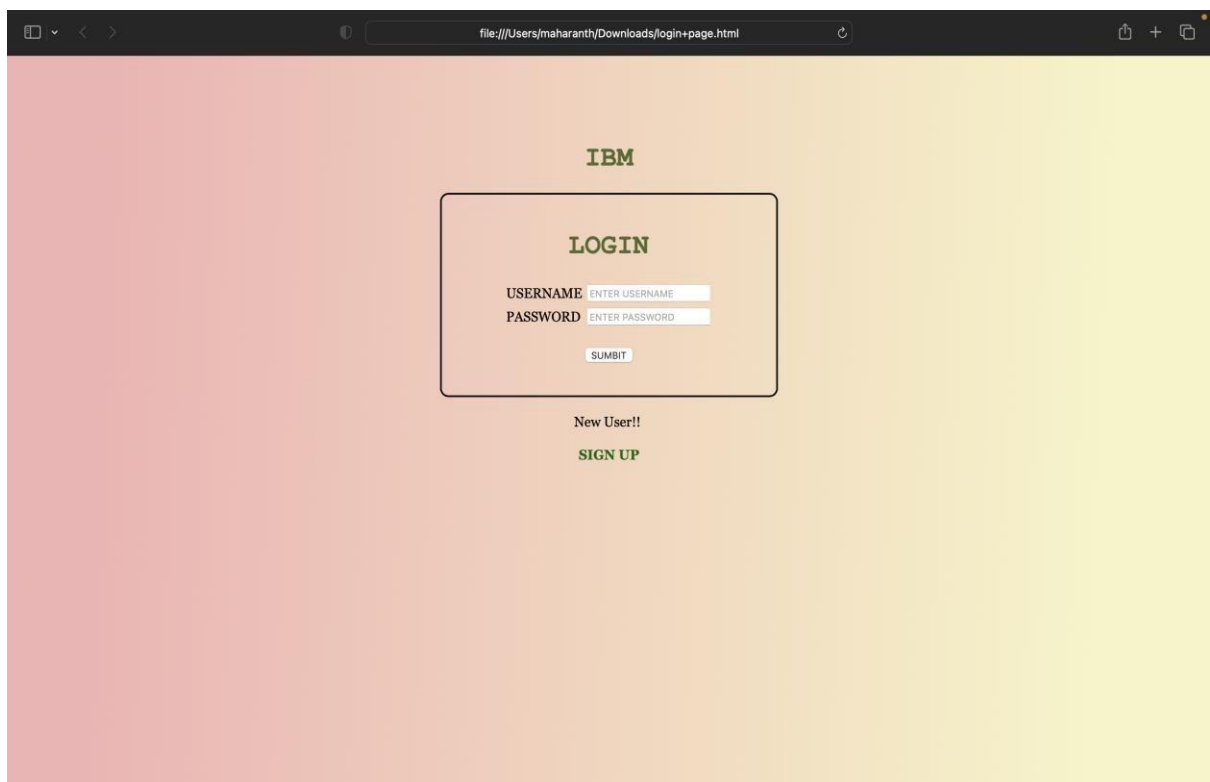




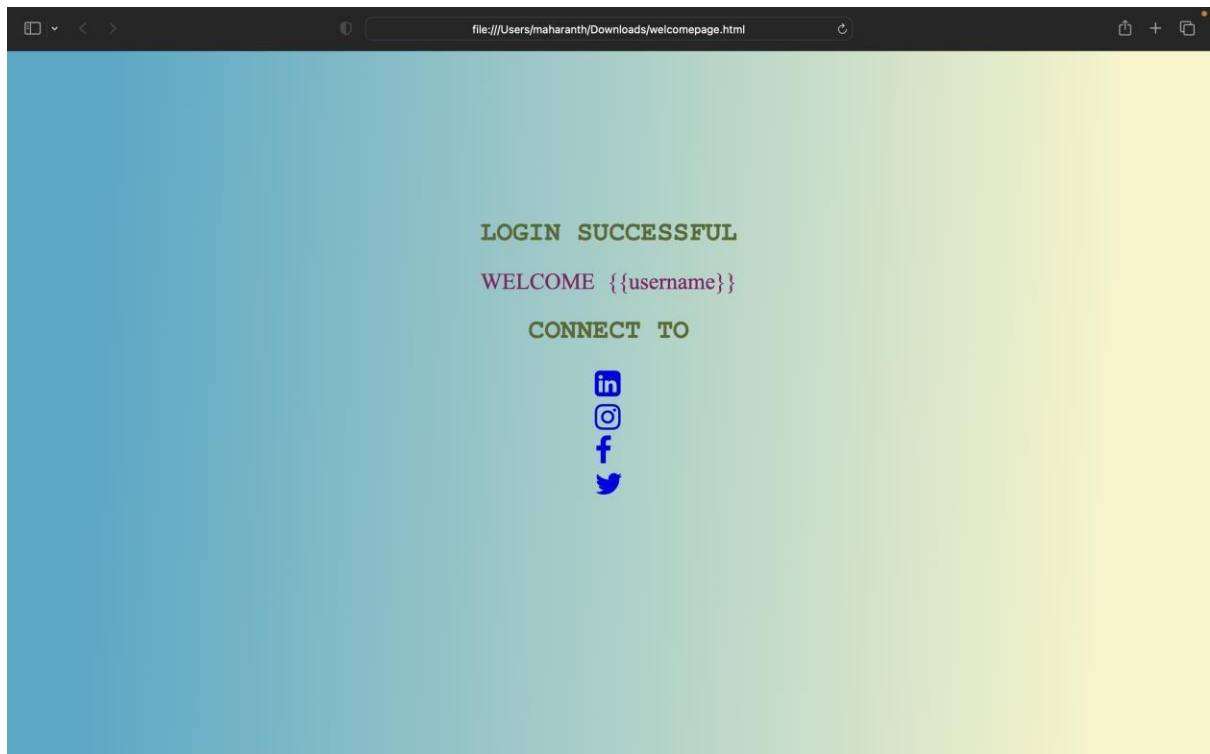
8.2 USER ACCEPTANCE TESTING



A screenshot of a web browser displaying a registration form. The browser's address bar shows the file path: `file:///Users/maharanth/Downloads/Registration.html`. The form is centered on a background with a blue-to-yellow gradient. It features a title "REGISTRATION FORM" in bold, green, uppercase letters. Below the title are four input fields, each with a label in bold, green, uppercase letters and a placeholder text in a smaller font: "USERNAME" (placeholder: "Enter Username"), "EMAIL ID" (placeholder: "Enter email_id"), "PHONE NUMBER" (placeholder: "Enter PHONE Number"), and "PASSWORD" (placeholder: "Enter PASSWORD"). A "Submit" button is located below the password field.



A screenshot of a web browser displaying a login page. The browser's address bar shows the file path: `file:///Users/maharanth/Downloads/login+page.html`. The page has a pink-to-yellow gradient background. At the top center is the heading "IBM" in bold, green, uppercase letters. Below it is a rounded rectangular box containing the title "LOGIN" in bold, green, uppercase letters. Inside this box are two input fields with labels in bold, green, uppercase letters and placeholder text: "USERNAME" (placeholder: "ENTER USERNAME") and "PASSWORD" (placeholder: "ENTER PASSWORD"). A "SUMBIT" button is located below the password field. Below the login box, the text "New User!!" is displayed, followed by a "SIGN UP" link in bold, green, uppercase letters.



10. ADVANTAGES AND DISADVANTAGES

- This app can be accessed anywhere and anytime, So that the user can view the news
- Its ad free
- The news is only based on the API
- It may contain some unwanted content but we don't have control over it
- The user can bookmark their favourite news.

11. CONCLUSION

Thus we have developed a full stack application based on the plans and within the given time. We have tested the application in both desktop and mobile and it worked well, Overall it was a great experience.

12. FUTURE SCOPE

In future we may integrate our own news API instead of third party APIs and may develop a mobile native application so that it can be used in both android and ios.