

# **NEWS TRACKERAPPLICATION**

USING CLOUD

*A Project report submitted in partial fulfilment of 7<sup>th</sup> semester in degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by*

Team ID: PNT2022TMID51975

Saranya R                      962319104079

Namit A R                     962319104060

Nivethitha M                962319104064

Venkadesh S R              962319104099



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY**

ANNA UNIVERSITY: CHENNAI 600025

**NOV-2022**

**AMRITA COLLEGE OF ENGINEERING AND  
TECHNOLOGY, ERACHAKULAM**



**BONAFIDE CERTIFICATE**

Certified that this project report "NEWS TRACKER APPLICATION" is the bonafiderecord work done by **SARANYA R(962319104079)**, **NAMIT A R (962319104060)**, **NIVETHITHA(962319104064)**, **VENKADESH S R (962319104099)** for **IBM- NALAIYATHIRAN** in **VII** semester of **B.E.**, degree course in **ComputerScience and Engineering** branch during the academic year of 2022 - 2023.

**Staff-In charge**

**Evaluator**

**HEAD OF THE DEPARTMENT**

## ACKNOWLEDGEMENT

We express our breathless thanks to our **Dr.T Kannan, M.E., Ph.D.**, the principal, Amrita College of Engineering and Technology, Erachakulam, for giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department (i/c) **Dr. P M Siva Raja M.Tech., Ph.D.**, for giving us valuable suggestion and help towards us throughout this Project.

We are highly grateful to thank our Project coordinator **Mrs.C Karpagavalli** and our Project Evaluator **Mrs.S L Jothi Lakshmi** Department of Computer Science and Engineering, Amrita College of Engineering and technology, for the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral support and suggestions to complete our Project work successfully

Finally our acknowledgement goes to our parents, sisters and friends those who had extended their excellent support and ideas to make our Project a pledge one.

Saranya R  
Namit A R  
Nivethitha M  
Venkadesh S R

## **ABSTRACT**

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.

“Newsapi” provides API that returns JSON (JavaScript Object Notation) metadata for headlines and articles live all around the world at any time . In this app we will be using this API for our better experience. Even after using this API it is possible that we can't reach maximum output of resources for that we can use Admin panel where admin or writers can add news of their own, manage and delete it. This contain will have separate database and access bar which will be labelled as “local news”. A module can be implemented where user will have different local news based on his location.

## **TABLE OF FIGURES**

### **1. INTRODUCTION**

1.1. Project Overview

1.2. Purpose

### **2. LITERATURE SURVEY**

2.1. Existing problem

2.2. References

2.3. Problem StatementDefinition

### **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 DeliverySchedule

6.3. Reports from JIRA

### **7. CODING & SOLUTIONING**

7.1. Feature 1

7.2. Feature 2Database Schema (if Applicable)

## 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

# News TrackerApplication

## INTRODUCTION

Mobile app ecosystems are transforming patterns of news consumption. Until quite recently, reading the news was a niche use for smartphones [12], mostly for when users were ‘on the go’; now however, two in every three users of mobile devices in the US regularly access news and as many as one in five read in-depth news articles daily [2]; a similar picture is found in the UK [1]. This growth in mobile news access continues the migration of news consumers to the Internet.

## Project overview

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

## Purpose

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.

# LITERATURE SURVEY

## Existing problem

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.

### Android News App

**Author:** Brijesh Joshi , Nehal Patel

**Journal:** International Journal of Applied Engineering Research ISSN0973-4562Volume 13, Number 11

**Published Year:** 2018

As world's technology is rapidly growing we has fast connection and networktoinstantly connect to other person. Day to day use in mobile, tablets andlaptopisincreasing, most of the people already have this facilities. In this fast and information oriented world we need to stay updated with every incidents andnewstoo. This News app is android mobile application where user have access tolatest 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

## Exploring mobile news reading interactions for news app personalization

**Author:** Marios Constantinides , Dowel

**Journal:** International Journal of Applied Engineering Research ISSN0973-4562Volume 13, Number 11

**Published Year:** August 2015

As news is increasingly accessed on smartphones and tablets, the needforpersonalising news app interactions is apparent. We report a series of threestudiesaddressing key issues in the development of adaptive news app interfaces. Wefirstsurveyed users' news reading preferences and behaviours; analysis revealedthreeprimary types of reader. We then implemented and deployed an Androidnewsappthat logs users' interactions with the app. We used the logs to train a classifier and showed that it is able to reliably recognise a user according to their reader type. Finally we evaluated alternative, adaptive user interfaces for each reader type.



## **Topic Detection and Tracking in NewsArticle**

**Author:** Sagar Patel, Sandip Patel, Sanket Suthar

**Journal:** DOI:10.1007/978-3-319-63645-0\_48

**Published Year:** March 2017

They have presented an idea in this paper for detecting and tracking topics from news articles. Topic detection and tracking are used in text mining process. From data which are unstructured in text mining we extract previously unknown and useful information. The main purpose of this paper is to identify and follow tasks occurred in different news sources. We are going to use agglomeration clustering based on average linkage for detecting the topics, calculate the similarity of topics using cosine similarity and KNN classifier for tracking the topics.

## **References**

Like

articles, websites, blogs

<https://www.researchgate.net/>.

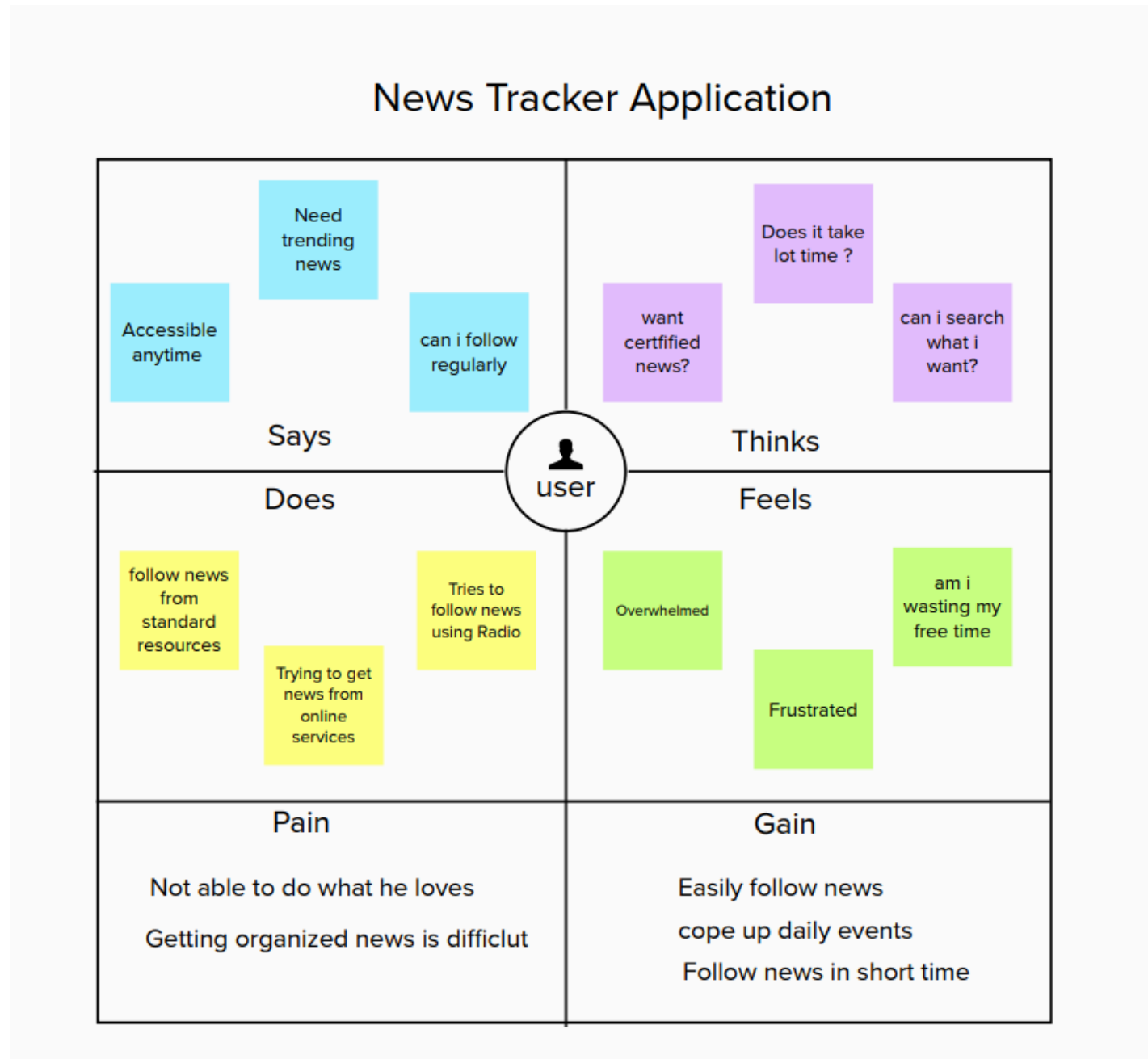
<https://techvidvan.com/tutorials/android-news-app-project-source-code/>

## **Problem Statement Definition**

1. The user needs a way to organise the news on daily basis so that he can cope up with daily events in his tight schedule.
2. The user wants to read news only about particular topics so that he can be informed about his interest.
3. The user wants to get informed from only certified news outlets.
1. The user needs a way to search about the news on topics he wants to.
2. The user wants to know the news about his surrounding using GPS location.

## IDEATION & PROPOSED SOLUTION

### Empathy Map Canvas



# Ideation & Brainstorming

## Brainstorm, Idea Listing and Grouping

NEWS APP WITHOUT SPAM	FAKE NEWS	GAINING WORLD KNOWLEDGE
IDENTIFY RELEVANT INFORMATION	INSTANT KNOWING OF HAPPENINGS AT OUR AREAS	READERS CAN VIEW WHAT NEWS THEY WANT TO KNOW
CURRENT AFFAIRS	REAL NEWS WITH REAL TIME UPDATE	OBSERVE AND WITNESS TRUE EVENTS

REPORTING AND ANALYTICS	INCREASE BRAND AWARENESS	CONNECTING WITH PEOPLE ALL OVER THE WORLD
BETTER FEEDBACK FROM CUSTOMERS	AUDIO-VIDEO ELEMENT	TARGETING DIFFICULT
LOW MARKET PRICE FOR COMMERCIAL THING	AVAILABILITY AND CONNECTIVITY	SECURITY CONCERNS

ALL THE INFORMATIN UNDER ONE ROOF	NEGATIVITY	LIVE VIDEOS
SOME CONTENT ARE INAPPROPRIATE TO CHIDREN	IMMEDIATE KNOWLEDGE OF BREAKING EVENTS	DAILY NEWS UPDATED
TOP NEWS STORIES OF THE DAY	MEDICAL NEWS AND HEALTH NEWS	BREAKING HEADLINES

OBSERVES AND WITNESSESS EVENT	MONITORS POWERFUL PERSONS	SHARING OF INFORMATION
AWARENESS	TECHNOLOGY DEVELOPMENT	INTERESTING FACTS
DAILY LIVES	ECONOMIC CRISIS	MORE ENTERTAINMENT

## Proposed Solution

S.NO	PARAMETERS	DESCRIPTION
1.	Problem statement (problems to be solved)	Encounter several events of local and national significance every day, And were ly on NEWS to inform us about them.
2.	Idea/solution description	One platform to read all of his/her favourite local news, dependence on search engines and social media.  We deliver news in the form of text and images
3.	Novelty/Uniqueness	We deliver Preferred news to people.
4.	Social Impact/Customer satisfaction	Since the news is delivered according to what the customer needs, his/her needs are satisfied on the basis of news gathering
5.	Business Model (Revenue)	Advertisement and premiumaccount to enjoyad-free News
6.	Scalability of the solution	We can deliver news as videos, and spot different content from different genres that point to a same factor and group them together

## Problem Solution fit

<p><b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span></p> <p>For young people to old people</p> <p>For poor persons to rich persons</p> <p>Also for village people to city people</p> <p>For working professionals to jobless persons</p>	<p><b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span></p> <p>No network, Provide Download option</p> <p>No well organized content</p> <p>No related and interesting and educating content</p> <p>Click Bait(Topic and content are not related)</p> <p>Annoying user interface</p> <p>No customization option</p>	<p><b>5. AVAILABLE SOLUTIONS</b> <span>AS</span></p> <p>Chat bot that solves user queries</p> <p>User will customize what the content to read and search</p> <p>Will Providing quick access to favorite topics</p> <p>User friendly interface, avoiding misleading ads</p> <p>Prioritize news according to user interest and location.</p>
<p><b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span></p> <p>Reading an unwanted and irrelevant and repeated content</p> <p>Very Bad user interface</p> <p>Searching related news</p> <p>Unwanted Ads contents and unorganized contents</p> <p>user unable to customize news content</p> <p>Using internet for previously watched content</p> <p>Providing dark mode</p>	<p><b>9. PROBLEM ROOT CAUSE</b> <span>RC</span></p> <p>No user customization. This leads to unorganized and uninterested news.</p> <p>No search bars leads frustration to search contents</p> <p>No service to complain(Chat bot solves queries.)</p> <p>No download option user may not have internet.</p> <p>Light mode may not good for eye.</p> <p>User interface needs to be attractive and easy to use or it make user to lose interest on app</p> <p>Provide notification which related to user wish</p>	<p><b>7. BEHAVIOUR</b> <span>BF</span></p> <p>User searching news and wasting time on it</p> <p>User gets frustrated while using bad user interface</p> <p>Misleading ads or topics wasting user time and confuse user</p> <p>User need all types of content but needs their favorite content to be prioritized.</p> <p>User may feel stressed eye.</p> <p>User may avoid notification if it is not related.</p>
<p><b>3. TRIGGERS</b> <span>TR</span></p> <p>People expecting about latest news</p> <p>When a news goes viral</p> <p>Need report about weather, market, sports and etc.</p> <p><b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span></p> <p>While reading irrelevant content make to</p> <p>Feels waste of time to read.</p> <p>Feels getting lack of information from contents</p>	<p><b>10. YOUR SOLUTION</b> <span>AL</span></p> <p>Providing search bars and content customization tiles</p> <p>Enabling download options and save or pin post options</p> <p>Providing Chat bot</p> <p>Providing Dark Mode</p> <p>Providing like, comments, tag, polling options to develop to develop user interface further more.</p> <p>User can control their notification. They can select content which they need to notified.</p>	<p><b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span></p> <p>8.1 ONLINE</p> <p>User can customize their news according to their interest.</p> <p>User can interact with community feed and user can report any queries</p> <p>8.2 OFFLINE</p> <p>User can save post and then read it for later</p> <p>User can download post and can share it to other people.</p>

## REQUIREMENT ANALYSIS

### Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via EmailConfirmation via OTP
FR-3	User Login	User using his /her device to login to theapp/website for reading news
FR-4	User search	For getting news he/she must need to search thenews in the given search bar, Which will collect the news according to his searchfrom the onlinesources like blog, e-news paper,Articles etc
FR-5	User Reads	Gathered news will be shown to the user in shortlines/ summary manner. Along with news Pictures, hyperlinks for the blog,article Or original publisher of the news is given
FR-6	User features	User can have lot of features like Like and shareand Post features.

## Non-functional Requirements:

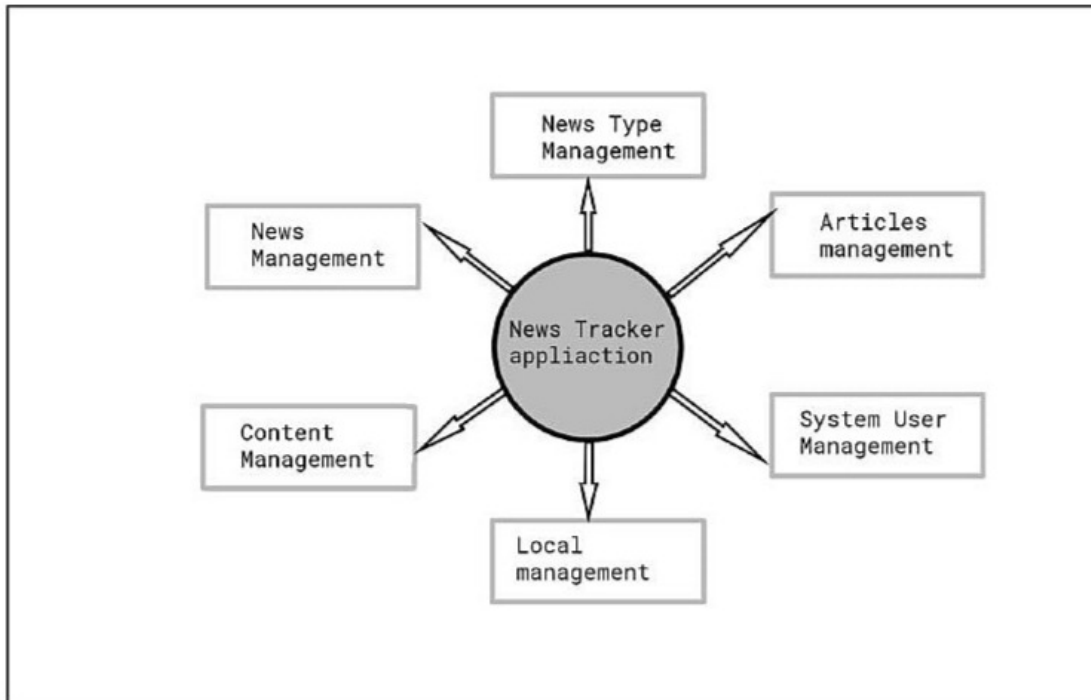
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	For getting news he/she must need to search the news in the given search bar, Which will collect the news according to his search from the online sources like blog, e-newspaper, Articles etc. Gathered news will be shown to the user in shortlines/ summary manner. Along with news Pictures, hyperlinks for the blog, article Or original publisher of the news is given
NFR-2	<b>Security</b>	User's data for the Registration and data collected from his activity are maintained in high security and must be used for only product improvement. Disclosing of his/her detail is considered as a crime hence they can take any legal actions against the service provider.
NFR-3	<b>Reliability</b>	News gathered and presented to the user are from only authorized publishers, and the sources are authentic. User can rely on this app for original news .
NFR-4	<b>Performance</b>	24*7 service with best cloud service, There will be no performance issue to the users.
NFR-5	<b>Availability</b>	Can download from Playstore and online websites.
NFR-6	<b>Scalability</b>	Hence it is cloud based system Scalability will be no issue because Scalability is the prior feature of the cloud based services.

## PROJECT DESIGN

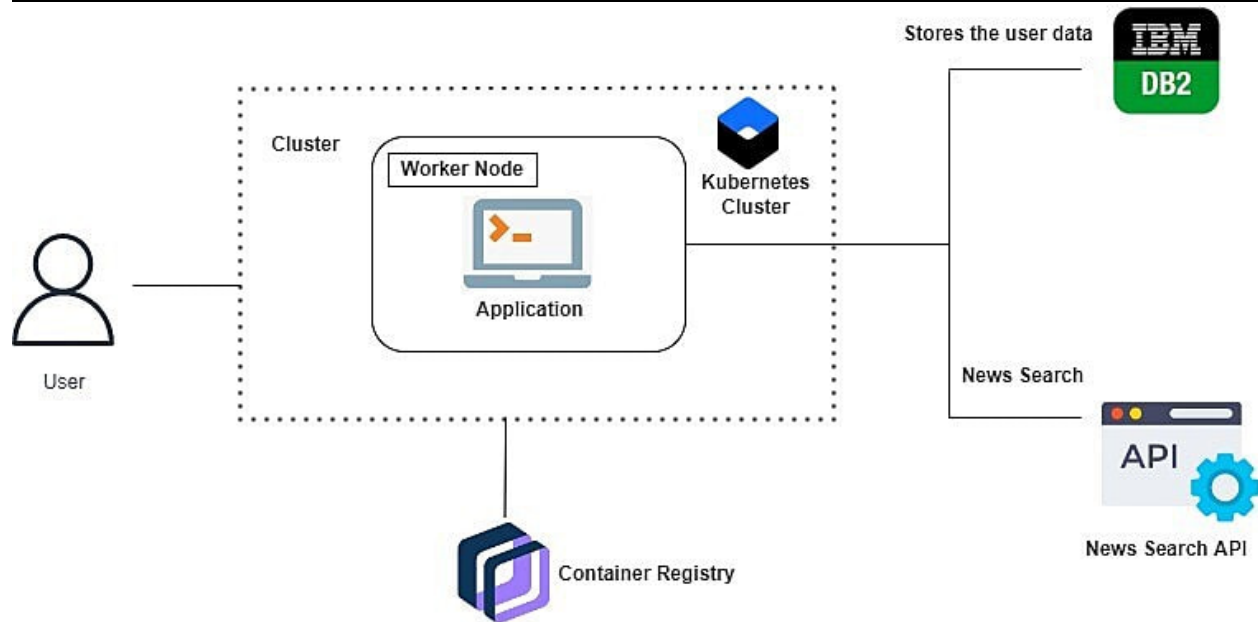
### Data Flow Diagrams

Example: DFD Level 0 (Industry Standard)





## Solution & Technical Architecture



## 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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		High	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	Log in	USN-6	As a user, I can access the application through website	I can access my account / dashboard	High	
Customer Care Executive	24x7 service	USN-7	As a User, I may need help in using the app/website	Any help need	High	
Administrator	Owner/Chief executive	-	He/she will access the datas in the app	Product behaviour/Improvement purposes	Low	

## PROJECT PLANNING & SCHEDULING

### Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	2 Sept 2022
Prepare Empathy Map	Prepare Empathy MapCanvas to capture the user Pains & Gains, Prepare list of problem statements	10 Sept 2022

Ideation Brain Storming	List the by organising the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	15 Sept 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 Sept 2022

Problem Solution Fit	Prepare problem solution fit document	28 Sept 2022
Solution Architecture	Prepare solution architecture document.	5 October 2022
Customer Journey	Prepare the customer journey maps to	7 October 2022

Solution Requirement	Prepare the functional requirement document	8 October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	10 October 2022
Technology Architecture	Prepare the technology architecture diagram.	15 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	26 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	17 November 2022

## **Sprint DeliverySchedule**

## Reports from JIRA

## CODING & SOLUTIONING

Forgot

password-

Forpass.ht

ml

```
<!doctypehtml>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

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

```
<meta name="description" content="">
```

```
<meta name="author" content="Mark Otto,Jacob Thornton, and Bootstrap contributors">
```

```
<meta name="generator" content="Hugo 0.84.0">
```

```
<title>Sign In</title>
```

```
<link rel="canonical" href="https://getbootstrap.com/docs/5.0/examples/sign-in/">
```

```
<link href="https://getbootstrap.com/docs/5.0/assets/css/docs.css" rel="stylesheet">
```

<!-- Bootstrap core CSS -->

<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"  
rel="stylesheet" integrity="sha384-  
iYQeCzEYFbKjA/T2uDLTPkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fz  
T" crossorigin="anonymous">

<!-- Favicons -->

<link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-touch-  
icon.png" sizes="180x180">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-32x32.png"  
sizes="32x32" type="image/png">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-16x16.png"  
sizes="16x16" type="image/png">

<link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">

<link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-pinned-  
tab.svg" color="#7952b3">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">

<meta name="theme-color" content="#7952b3">

<style>

.bd-placeholder-  
img { font-size:  
1.125rem; text-  
anchor: middle;  
-webkit-user-select: none;  
-moz-user-select:  
none;user-select:

```
    none;
}

@media (min-width: 768px){
    .bd-placeholder-
    img-lg {font-size:
    3.5rem;
    }
}
</style>
```

```
<!-- Custom styles for this template -->
<link href="static/sheets/signin.css" rel="stylesheet">
<link href="static/sheets/colors.css" rel="stylesheet">
</head>
<body class="text-center">
    <nav class="navbar navbar-dark bg-darkfixed-top">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">News Tracker</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="offcanvas" data-bs-
target="#offcanvasDarkNavbar" aria-controls="offcanvasDarkNavbar">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="offcanvas offcanvas-end text-bg-dark"
tabindex="-1" id="offcanvasDarkNavbar" aria-
labelledby="offcanvasDarkNavbarLabel">
```

```
<div class="offcanvas-header">

  <h5 class="offcanvas-title" id="offcanvasDarkNavbarLabel">Profile</h5>

  <button type="button" class="btn-close btn-close-white"
data-bs-dismiss="offcanvas" aria-label="Close"></button>

</div>

<div class="offcanvas-body">

  <ul class="navbar-nav justify-content-end flex-grow-1 pe-3">

    <li class="nav-item">

      <a class="nav-link active" aria-current="page" href="home.html">Home</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="base.html">Fetch News</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="about.html">About Us</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="signin.html">Sign In</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="signup.html">Sign Up</a>

    </li>

  </div>

</div>
```



```

    </div>

</nav>

<main class="form-signin">

    <form action="{ { url_for('getUser') } }" method="POST">

        <h1 class="h3 mb-3 fw-normal white">Change your password</h1>


        <div class="form-floating">

            <input type="email" class="form-control" id="floatingInput" name =
"uname"placeholder="name@example.com">

            <label for="floatingInput">Email address</label>

        </div>

        <br>

        <button class="w-100 btn btn-lg btn-warning btn-outline-warning" type="submit"><a
href="static/templates/changepass.html">Change Password</a></button><br><br>

    </form>

</main>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C
8" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@docsearch/js@3"></script>

<script src="https://cdn.jsdelivr.net/npm/@stackblitz/sdk@1/bundles/sdk.umd.js"></script>

<script src="/docs/5.2/assets/js/docs.min.js"></script>

```

```
<script>
```

```
document.querySelectorAll('.btn-edit').forEach(btn
=> {btn.addEventListener('click', event => {
    const    htmlSnippet    =    event.target.closest('.bd-code-
snippet').querySelector('.bd-example').innerHTML

    const classes = Array.from(event.target.closest('.bd-code-
snippet').querySelector('.bd-example').classList).join(' ')

    const    jsSnippet    =    event.target.closest('.bd-code-
snippet').querySelector('.btn-edit').getAttribute('data-sb-js-snippet')
    StackBlitzSDK.openBootstrapSnippet(htmlSnippet, jsSnippet, classes)
    })
    })

StackBlitzSDK.openBootstrapSnippet = (htmlSnippet, jsSnippet, classes)
=> {const markup =`<!doctype html>
<html lang="en">
<head>

<meta charset="utf-8">

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

<link
```





```

    \u003cdiv class=\u0022alert alert-\${type}\ alert-\ dismissible\u0022
role=\u0022alert\u0022\u003e`,\n
\u003cdiv\u003e\${message}\u003cdiv\u003e`,\n
\u003cbutton type=\u0022button\u0022 class=\u0022btn-\u0022close\u0022 data-\u0022bs-\u0022dismiss=\u0022alert\u0022
aria-\u0022label=\u0022Close\u0022\u003e\u003cbutton\u003e\u0027,\n
\u0027\u003cdiv\u003e\u0027\n
\].join(\u0027\u0027)\n\n alertPlaceholder.append(wrapper)\n
}\n\n
if (alertTrigger) {\n alertTrigger.addEventListener(\u0027click\u0027,\n
)\n
=\u003e\n
{\n appendAlert(\u0027Nice, you triggered this alert
message!\u0027, \u0027success\u0027)\n
})\n
}\n
\u0027\u0027 Checks \u0026Radios\n
\u0027\u0027 Indeterminate checkbox
example in docs and StackBlitz\n
document.querySelectorAll(\u0027.bd-\u0027example-\u0027indeterminate
[type=\u0022checkbox\u0022]\u0027)\n
.forEach((checkbox =\u003e{\n
if
(checkbox.id.includes(\u0027Indeterminate\u0027)) {\n
checkbox.indeterminate =true\n
}\n
})\n
)\n
\u0027\u0027 Links\n
\u0027\u0027 Disable emptylinks in docs
examplesonly\n
document.querySelectorAll(\u0027.bd-\u0027content
[href=\u0022#\u0022]\u0027)\n
.forEach((link =\u003e{\n
link.addEventListener(\u0027click\u0027, event =\u003e
{\n
event.preventDefault()\n
})\n
})\n
)\n
\u0027\u0027 \u0027\u0027 Modal\n
\u0027\u0027 \u0027\u0027 Modal
\u0027\u0027 Varying modal content\u0027 example in docs and StackBlitz\n
const exampleModal
= document.getElementById(\u0027exampleModal\u0027)\n
if (exampleModal) {\n
exampleModal.addEventListener(\u0027show.bs.modal\u0027, event =\u003e {\n
\u0027\u0027 Button that triggered the modal\n
const button = event.relatedTarget\n
\u0027\u0027 Extract info
from data-\u0022 attributes\n
const recipient=
button.getAttribute(\u0027data-\u0022 whatever\u0027)\n
\u0027\u0027 Update the
modal\u0027s content.\n
const modalTitle =
exampleModal.querySelector(\u0027.modal-\u0027title\u0027)\n
const modalBodyInput =
exampleModal.querySelector(\u0027.modal-\u0027body
input\u0027)\n
modalTitle.textContent = `New message to \${recipient}`\n
modalBodyInput.value = recipient\n
}\n
)\n
\u0027\u0027 \u0027\u0027 Offcanvas\n
\u0027\u0027 \u0027\u0027 Offcanvas

```

```

components\u0027    example    in    docs    only\n
document\\.querySelectorAll\\(\u0027\\.bd\\-example\\-offcanvas    \\.offcanvas\u0027\\)\n
if
\\(myOffcanvas\\)    \\{\n
    myOffcanvas\\.forEach\\(offcanvas =\u003e \\{\n
offcanvas\\.addEventListener\\(\u0027show\\.bs\\.offcanvas\u0027,    event    =\u003e \\{\n
event\\.preventDefault\\(\\)\n    \\}, false\\)\n    \\})\n    \\}\n\\}\n\\}\n : null

```

```
const
```

```
project    =
```

```
{files: {
```

```
    'index.html': markup,
```

```
    'index.js':
```

```
        jsSnippetContent
```

```
},
```

```
title: 'Bootstrap Example',
```

```
description: `Officialexample from ${window.location.href}`,
```

```
template: jsSnippet ? 'javascript' : 'html',
```

```
tags: ['bootstrap']
```

```
}
```

```
StackBlitzSDK.openProject(project, { openFile: 'index.html' })
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

## TESTING

### Test Cases

Test case

Test case	feature	component	Test scenario	Expected result	Actual result	status	comments	bug	Executed by
Sign in	Functional	Login page	Verify user can see the sign in option	can visible	Yes visible	pass	successful	-	Abishek
Sign up	Functional	Login page	Verify user has the option to sign up	Can visible	Yes visible	pass	Successful	-	Abishek
Forgot password	Functional	Login page	Verify user has the option to forgot password	Yes the option is available	Option is available	pass	Successful	-	Manikandan

Fetch news	Functional	Home page	Verify user can get the news	News will be feed to the app	404 error	Fail	unsuccessful	App integration problem	
Types of news available in the fetch news page	Functional	Fetch news	Types of news available	Weather, Sport, Economy.	Hover buttons will be shown.	Yes	successful	-	namit

**Result:**

News tracker application using cloud is developed and executed at the level of completed progress .

**ADVANTAGES & DISADVANTAGES****Advantages:-**

- In this app news is already categorization.
- Easily accessible and portable.
- Better user experience.
- Apps help you convert visitors to loyal readers.
- Minute by minute updates of news.
- This app helps you to get local news updates instantly.
- To explore and discover trending news and topics.
- News feeds for you based on your interest.

**Disadvantages:-**

- ✓ Some apps demand premium subscription from user.
- ✓ Occurrence of Advertisement disturbs the user.
- ✓ Sometimes the news gives brief information.
- ✓ Prevalence of fake and uncertain news can confuse the user and lead to misconception.
- ✓ Fake news may mislead the readers.



## **CONCLUSION**

We explored the feasibility of recognising patterns of news reading interactions and evaluated three adaptive interface designs for different news reader types. We show that from their interaction log, a specific user can be recognised as one of three kinds. The reader types emerging from the online survey are well defined and distinct. The evaluation of the three variant interfaces suggests that different news reader types need different user interfaces. We have demonstrated a method for monitoring users' news reading behaviour and inferring news reader type from it. In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

## **FUTURE SCOPE**

In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps

## APPENDIX

### SOURCE CODE

main.py

```
from app import app

if __name__ == "__main__":
    app.run(debug=True)
```

Init.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_bcrypt import Bcrypt
from flask_login import UserMixin
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt

app = Flask(__name__)

bcrypt = Bcrypt(app)
app.config["SQLALCHEMY_DATABASE_URI"] = 'sqlite:///database.db'
app.config["SECRET_KEY"] = 'thisisasecretkey'
db = SQLAlchemy(app)
```

```

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key = True)
    username = db.Column(db.String(20), nullable = False, unique = True)
    password = db.Column(db.String(20), nullable = False)

class RegisterForm(FlaskForm):
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Username"})
    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})
    submit = SubmitField('Register')
    def validate_username(self, username):
        existing_user_username = User.query.filter_by(
            username=username.data).first()
        if existing_user_username:
            raise ValidationError(
                'That username already exists. Please choose a different one.')

class LoginForm(FlaskForm):
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Username"})
    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})
    submit = SubmitField('Login')

from app import views

```

## Config.py

```

class Config:

    NEWS_BASE_URL_SOURCES = 'https://newsapi.org/v2/top-headlines/sources?apiKey={}'
    NEWS_BASE_EVERYTHING_URL = 'https://newsapi.org/v2/everything?domains={}&apiKey={}'
    NEWS_BASE_HEADLINES_URL = 'https://newsapi.org/v2/top-headlines?country=in&apiKey={}'
    NEWS_BASE_SOURCE = 'https://newsapi.org/v2/top-headlines?sources={}&apiKey={}'

```

```
API_KEY = "6b9692ac53e94863ba34ba3bc8e727c4"

class ProdConfig(Config):
    Pass

class DevConfig(Config):
    DEBUG = True
    config_options = {
        'development': DevConfig,
        'production': ProdConfig
    }
```

## Models.py

```
class Sources:
    def __init__(self, name, description, url):
        self.name=name,
        self.description=description
        self.url=url

class Articles:
    """Define article model"""
    def __init__(self, source, author, title, description, url, urlToImage, publishedAt):
        self.source = source
        self.author = author
        self.title = title
        self.description = description
        self.url = url
        self.urlToImage = urlToImage
        self.publishedAt = publishedAt
```

## Request.py

```
from app.models import Articles
from app.models import Sources
from newsapi import NewsApiClient
from app.config import Config
import urllib.request,json
```

```

api_key=None
base_url=None
base_url_for_everything=None
base_url_top_headlines=None
base_source_list=None
all_types_articles = []

def news_sorter(all_articles):

    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):

        article = all_articles[i]
        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])
        article_object = Articles(source, title, desc, author, img, p_date, url)
        all_types_articles.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)

    return contents

def publishedArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    news= []

    #todays news

```

```
get_articles = newsapi.get_everything(sources= 'cnn, reuters, cnbc, the-verge, gizmodo, the-next-web,
techradar, recode, ars-technica')
all_articles = get_articles['articles']

output = news_sorter(all_articles)
news.append(output)

#top headlines

get_articles = newsapi.get_top_headlines(sources= 'cnn, reuters, cnbc, techcrunch, the-verge, gizmodo, the-
next-web, techradar, recode, ars-technica')
all_articles = get_articles['articles']

output = news_sorter(all_articles)
news.append(output)

#business articles

get_articles = newsapi.get_top_headlines(category='business')
all_articles = get_articles['articles']

output = news_sorter(all_articles)
news.append(output)

#tech articles

get_articles = newsapi.get_top_headlines(category='technology')
all_articles = get_articles['articles']

output = news_sorter(all_articles)
news.append(output)

#entertainment_articles

get_articles = newsapi.get_top_headlines(category='entertainment')
all_articles = get_articles['articles']

output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#science_articles
```

```
science_articles = newsapi.get_top_headlines(category='science')
```

```
all_articles = science_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#sport articles
```

```
sport_articles = newsapi.get_top_headlines(category='sports')
```

```
all_articles = sport_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#health articles
```

```
get_articles = newsapi.get_top_headlines(category='health')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
    return news
```

```
def topHeadlines():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    top_headlines = newsapi.get_top_headlines(sources= 'cnn, reuters, cnbc, techcrunch, the-verge, gizmodo,  
    the-next-web, techradar, recode, ars-technica')
```

```
    all_headlines = top_headlines['articles']
```

```
    return news_sorter(all_headlines)
```

```
def randomArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    random_articles = newsapi.get_everything(sources= 'the-verge, gizmodo, the-next-web, recode, ars-
```

```
technica')  
    all_articles = random_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def businessArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)  
    business_articles = newsapi.get_top_headlines(category='business')  
    all_articles = business_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def techArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)  
    tech_articles = newsapi.get_top_headlines(category='technology')  
    all_articles = tech_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def entArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)  
    ent_articles = newsapi.get_top_headlines(category='entertainment')  
    all_articles = ent_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def scienceArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)  
    science_articles = newsapi.get_top_headlines(category='science')  
    all_articles = science_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def sportArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)  
    sport_articles = newsapi.get_top_headlines(category='sports')  
    all_articles = sport_articles['articles']
```



```

        return news_sorter(all_articles)

def healthArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    health_articles = newsapi.get_top_headlines(category='health')
    all_articles = health_articles['articles']

    return news_sorter(all_articles)

def get_news_source():
    """
    Function that gets the json response to our url request
    """
    get_news_source_url = 'https://newsapi.org/v2/sources?apiKey=' + Config.API_KEY
    with urllib.request.urlopen(get_news_source_url) as url:
        get_news_source_data = url.read()
        get_news_source_response = json.loads(get_news_source_data)
        news_source_results = None

    if get_news_source_response['sources']:
        news_source_results_list = get_news_source_response['sources']
        news_source_results = process_sources(news_source_results_list)

    return news_source_results

def process_sources(source_list):
    """
    function that process the news articles and transform them to a list of objects
    """
    news_source_result = []
    for news_source_item in source_list:
        name = news_source_item.get('name')
        description = news_source_item.get('description')
        url = news_source_item.get('url')

    if name:
        news_source_object = Sources(name, description,url)
        news_source_result.append(news_source_object)
    return news_source_result

```

## Views.py

```
from app import app
import ibm_db

from flask import session, flash, redirect, render_template, request, url_for
from .request import (businessArticles, entArticles, get_news_source,
                      healthArticles, publishedArticles, randomArticles,
                      scienceArticles, sportArticles, techArticles,
                      topHeadlines)

conn = ibm_db.connect(
    'DATABASE=bludb;'
    'HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;'
    'PORT=30875;'
    'SECURITY=SSL;'
    'SSLServerCertificate=DigiCertGlobalRootCA.crt;'
    'UID=rzf07928;'
    'PWD=Gsmx9dW4PeOQMqze;', "", ""
)

global account

@app.route('/', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        # getting user data
        email = request.form.get('email')
        password = request.form.get('password')
        sql_check_query = "SELECT * FROM user WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql_check_query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

```

print(account)
if account:
    # email id exists
    # checking if the password is correct
    if not account['PASSWORD'] == password:
        flash('Invalid password', category='error')

    else:
        # user entered the correct password
        # redirecting the user to the dashboard
        session['user_id'] = account['EMAIL']
        return redirect(url_for('home'))

else:
    # email id does not exist in the database
    flash('Email invalid... Try Again', category='error')

return render_template('auth/login.html')

return render_template('auth/login.html')
# return render_template('login.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        # getting user data
        email = request.form.get('email')
        password = request.form.get('password')
        # checking: user already exists or not
        sql_check_query = "SELECT * FROM user WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql_check_query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)
        # email id does not exist in the database
        if not account:
            # inserting the data into the database
            sql_insert_query = "INSERT INTO user VALUES (?, ?)"

```



```
        health_articles = health_articles,
        account = session['user_id'])

@app.route('/headlines')
def headlines():
    headlines = topHeadlines()

    return render_template('headlines.html', headlines = headlines, account = session['user_id'])

@app.route('/articles')
def articles():
    random = randomArticles()

    return render_template('articles.html', random = random, account = session['user_id'])

@app.route('/sources')
def sources():
    newsSource = get_news_source()

    return render_template('sources.html', newsSource = newsSource, account = session['user_id'])

@app.route('/category/business')
def business():
    sources = businessArticles()
    return render_template('business.html', sources = sources, account = session['user_id'])

@app.route('/category/tech')
def tech():
    sources = techArticles()

    return render_template('tech.html', sources = sources, account = session['user_id'])

@app.route('/category/entertainment')
def entertainment():
    sources = entArticles()

    return render_template('entertainment.html', sources = sources, account = session['user_id'])

@app.route('/category/science')
```

```

def science():
    sources = scienceArticles()

    return render_template('science.html', sources = sources, account = session['user_id'])

@app.route('/category/sports')
def sports():
    sources = sportArticles()

    return render_template('sport.html', sources = sources, account = session['user_id'])

@app.route('/category/health')
def health():
    sources = healthArticles()

    return render_template('health.html', sources = sources, account = session['user_id'])

```

## HTML Templates Source Code

### Base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="Description" content="Enter your description here"/>
    <!-- Favicons -->
    <link href="{{ url_for('static', filename='/favicon.ico') }}" rel="icon">

    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Overpass">
    <link rel="stylesheet"

```

```

href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css"
integrity="sha512-
tS3S5qG0BlhnQROyJXvNjeEM4UpMXHrQfTGmbQ1gKmelCxlSEBUaxhRBj/EFTzpbP4RVSrPEikbmdJob
CvhE3g==" crossorigin="anonymous" referrerpolicy="no-referrer" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous"> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
<link rel="stylesheet" href="{{ url_for('static', filename='assets/css/style.css') }}">
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzpe5IDxibcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<title>Spot-on Times</title>
</head>
<body>
    {% block navbar %}
    <div class="container-fluid" style="background-color: #000;">
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top shadow-sm">
            <!-- Show this only on mobile to medium screens -->
            <a class="navbar-brand d-lg-none" href="{{ url_for('home') }}"><b class="lg"><span
class="logo">Spot-On</span> Times</b></a>

            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggle" aria-
controls="navbarToggle" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <!-- Use flexbox utility classes to change how the child elements are justified -->
            <div class="collapse navbar-collapse justify-content-between" id="navbarToggle">

                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('headlines') }}">Headlines</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('articles') }}">Articles</a>
                    </li>
                    <li class="nav-item">

```

```

        <a class="nav-link" href="{{ url_for('sources') }}">Sources</a>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
toggle="dropdown" aria-expanded="false">
            Categories
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
            <a class="dropdown-item" href="{{ url_for('business') }}">Business</a>
            <a class="dropdown-item" href="{{ url_for('tech') }}">Technology</a>
            <a class="dropdown-item" href="{{ url_for('entertainment') }
}">Entertainment</a>
            <a class="dropdown-item" href="{{ url_for('science') }}">Science</a>
            <a class="dropdown-item" href="{{ url_for('sports') }}">Sport</a>
            <a class="dropdown-item" href="{{ url_for('health') }}">Health</a>
        </div>
    </li>
</ul>

<!-- Show this only lg screens and up -->
    <a class="navbar-brand d-none d-lg-block" href="{{ url_for('home') }}"><b class="lg"><span
class="logo">Spot-On</span>Times</b></a>

    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="#">
                <svg class="svg" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24"
height="24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2A10.13 10.13 0 0 0 2
12a10 10 0 0 4 7.92V20h.1a9.7 9.7 0 0 0 11.8 0h.1v-.08A10 10 0 0 0 22 12 10.13 10.13 0 0 0 12 2zM8.07
18.93A3 3 0 0 1 11 16.57h2a3 3 0 0 1 2.93 2.36 7.75 7.75 0 0 1 7.86 0zm9.54-1.29A5 5 0 0 0 13 14.57h-2a5 5
0 0 0 -4.61 3.07A8 8 0 0 1 4 12a8.1 8.1 0 0 1 8 8.1 8.1 0 0 1 8 8 8 0 0 1 -2.39 5.64z"></path><path d="M12
6a3.91 3.91 0 0 0 -4 4 3.91 3.91 0 0 0 4 4 3.91 3.91 0 0 0 4 4 3.91 3.91 0 0 0 4 4zm0 6a1.91 1.91 0 0 1 -2 -2
1.91 1.91 0 0 1 2 -2 1.91 1.91 0 0 1 2 2 1.91 1.91 0 0 1 -2 2z"></path></svg>{{ account }}</a>
        </li>
    </ul>
</div>
</nav>
</div>
{% endblock %}

```



```

{% block content %}
{% endblock %}
{% block footer %}
<div class="container-fluid footer">
  <p class="text-center footer-text">
    © <span class="logo">Spot-On </span> Times
  </p>
</div>
{% endblock %}
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.1/umd/popper.min.js"></script>
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNjuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.6.0/js/bootstrap.min.js"></script>
</body>
</html>

```

## login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Form</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='assets/css/loginstyle.css') }}">
</head>
<body>
  <form method="POST">
    <input type="email" placeholder="email" name="email" required>
    <input type="password" placeholder="password" name="password" required>
    <input type="submit" value = "login">
    <span>Forgot <a href="">Password</a></span>
    <br>
    <br>
    <span>New User? <a href="{{ url_for('register') }}">Register</a></span>

```

```
</form>
</body>
</html>
```

## Register.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>Login Form</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='assets/css/loginstyle.css') }}">
</head>
<body>
  <form method="POST">
    <input type="email" placeholder="email" name="email" required>
    <input type="password" placeholder="password" name="password" required>
    <input type="submit" value = "Register">
    <br>
    <span>Already a user? <a href="{{ url_for('login') }}">Login</a></span>
  </form>
</body>
</html>
```

## Home.html

```
{% extends 'base.html' %}

{% block content %}
<div class="bg-dark">
  <br>
  <br>
  <br>
  <h3 class = >TODAYS NEWS</h3><br>
  <div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in todays_news %}
      <div class = "media-element">
```

```

        
    </div>
    <p>{{ source.name }}</p>
    <a href="{{ url }}">
        <p><b> {{ title }} </b></p>
    </a>
</div>
</div>
{% endfor %}
</div>
<br>
<h3 class = >TOP HEADLINES</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in top_headlines %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}
</div>
<br>
<h3 class = >BUSINESS</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in business_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}

```

```

</div>
<br>
<h3 class = >TECH NEWS</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in tech_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}
</div>
<br>
<h3 class = >ENTERTAINMENT</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in entertainment_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}
</div>
<br>
<h3 class = >SCIENCE ARTICLES</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in science_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
            </div>
        </div>
    {% endfor %}
</div>

```

```

        <a href="{{ url }}">
            <p><b> {{ title }} </b></p>
        </a>
    </div>
</div>
{% endfor %}
</div>
<br>
<h3 class = >SPORTS</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in sport_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}
</div>
<br>
<h3 class = >HEALTH NEWS</h3><br>
<div class="media-scroller snaps-inline">
    {% for source, title, desc, author, img, p_date, url in health_articles %}
        <div class = "media-element">
            
            <div>
                <p>{{ source.name }}</p>
                <a href="{{ url }}">
                    <p><b> {{ title }} </b></p>
                </a>
            </div>
        </div>
    {% endfor %}
</div>
<br>
</div>

```

```
{% endblock %}
```

## Article.html (This template is used for all the news Categories)

```
{% extends 'base.html' %}

{% block content %}
<div class="container-fluid landing">
  <div class="elevation-demo-container">
    {% for source, title, desc, author, img, p_date, url in catogory_name%}
    <div class="elevation-demo-surface mdc-elevation--z19">
      <div class="card mb-3 bg-dark text-white">
        
        <div class="card-body">
          <p class="card-text source">{{ source.name }}</p>
          <a href="{{ url }}">
            <h2 class="card-title"><b> {{ title }} </b></h2>
          </a>
          <p class="card-text">{{ desc }}</p>
          <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;><path d="M12 2C6.579 2 2
6.579 2 12s4.579 10 10 10-4.579 10-10s17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-
1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828
18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>
        </div>
        <div class="card-footer">
          <p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform:
;msFilter:;><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8-3.589 8-8 3.589-8-8-8zm0 14c-3.309 0-6-
2.691-6-6s2.691-6 6-6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293
5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p_date }}</p>
        </div>
      </div>
    </div>
  </div>
  {% endfor %}
</div>
{% endblock %}
```

# Styling

## Loginstyle.css

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  background-color: #191919;
}

form{
  width: 350px;
  padding: 40px;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background: #191919;
  text-align: center;
  box-shadow: 0 20px 18px 0 rgba(0, 0, 0, 1), 0 5px;
}

form h1{
  color: white;
  text-transform: uppercase;
  background: #191919;
  font-weight: 500;
}

form input[type = 'email'], form input[type = 'password']{
  border: none;
  background: none;
  display: block;
  margin: 20px auto;
  text-align: center;
  padding: 14px 10px;
```

```
width: 200px;
outline: none;
color: white;
border-bottom: 1px solid #FFFF00;
}

form input[type = 'submit']{
border: none;
background: none;
display: block;
margin: 45px auto 20px auto;
text-align: center;
border: 2px solid #FFFF00;
padding: 14px 40px;
outline: none;
color: white;
border-radius: 24px;
cursor: pointer;
}

form input[type = 'submit']:hover{
background-color: #FFFF00;
}

form span{
color: grey;
font-size: 12px;
}

form a{
font-size: 12px;
color: #FFFF00;
text-transform: none;
}
```

**Style.css**



```
@import "https://unpkg.com/open-props";
@import "https://unpkg.com/open-props/normalize.min.css";
@import url("https://fonts.googleapis.com/css?family=Roboto:100,300,400,400i,500,500i,700,900");
@import url("https://cdn.materialdesignicons.com/2.0.46/css/materialdesignicons.min.css");

*,*:after,:before{
  box-sizing: border-box;
  outline: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
/*Title Default*/
body{
  background: #343436;
  color: #777;
  font-family: Roboto;
}
h1{
  margin: 1em 0;
  border-bottom: 1px dashed;
  padding-bottom: 1em;
  font-weight: lighter;
  text-align: center;
}
.elevation-demo-container {
  display: -ms-flexbox;
  display: flex;
  margin: auto;
  -ms-flex: 1 1 auto;
  flex: 1 1 auto;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
  -ms-flex-pack: justify;
  justify-content: space-between;
  width: 100%;
```

```
}

.elevation-demo-surface {
  display: -ms-inline-flexbox;
  display: inline-flex;
  -ms-flex-pack: distribute;
  justify-content: space-around;
  min-height: 100px;
  min-width: 200px;
  margin: 15px;
  -ms-flex-align: center;
  align-items: center;
}

.mdc-elevation--z0 {
  -webkit-box-shadow: 0px 0px 0px 0px rgba(0, 0, 0, 0.2), 0px 0px 0px 0px rgba(0, 0, 0, 0.14), 0px 0px 0px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 0px 0px 0px rgba(0, 0, 0, 0.2), 0px 0px 0px 0px rgba(0, 0, 0, 0.14), 0px 0px 0px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z1 {
  -webkit-box-shadow: 0px 2px 1px -1px rgba(0, 0, 0, 0.2), 0px 1px 1px 0px rgba(0, 0, 0, 0.14), 0px 1px 3px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 2px 1px -1px rgba(0, 0, 0, 0.2), 0px 1px 1px 0px rgba(0, 0, 0, 0.14), 0px 1px 3px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z2 {
  -webkit-box-shadow: 0px 3px 1px -2px rgba(0, 0, 0, 0.2), 0px 2px 2px 0px rgba(0, 0, 0, 0.14), 0px 1px 5px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 3px 1px -2px rgba(0, 0, 0, 0.2), 0px 2px 2px 0px rgba(0, 0, 0, 0.14), 0px 1px 5px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z3 {
  -webkit-box-shadow: 0px 3px 3px -2px rgba(0, 0, 0, 0.2), 0px 3px 4px 0px rgba(0, 0, 0, 0.14), 0px 1px 8px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 3px 3px -2px rgba(0, 0, 0, 0.2), 0px 3px 4px 0px rgba(0, 0, 0, 0.14), 0px 1px 8px 0px rgba(0, 0, 0, 0.12);
}
```

```
}

.mdc-elevation--z4 {
  -webkit-box-shadow: 0px 2px 4px -1px rgba(0, 0, 0, 0.2), 0px 4px 5px 0px rgba(0, 0, 0, 0.14), 0px 1px 10px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 2px 4px -1px rgba(0, 0, 0, 0.2), 0px 4px 5px 0px rgba(0, 0, 0, 0.14), 0px 1px 10px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z5 {
  -webkit-box-shadow: 0px 3px 5px -1px rgba(0, 0, 0, 0.2), 0px 5px 8px 0px rgba(0, 0, 0, 0.14), 0px 1px 14px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 3px 5px -1px rgba(0, 0, 0, 0.2), 0px 5px 8px 0px rgba(0, 0, 0, 0.14), 0px 1px 14px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z6 {
  -webkit-box-shadow: 0px 3px 5px -1px rgba(0, 0, 0, 0.2), 0px 6px 10px 0px rgba(0, 0, 0, 0.14), 0px 1px 18px 0px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 3px 5px -1px rgba(0, 0, 0, 0.2), 0px 6px 10px 0px rgba(0, 0, 0, 0.14), 0px 1px 18px 0px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z7 {
  -webkit-box-shadow: 0px 4px 5px -2px rgba(0, 0, 0, 0.2), 0px 7px 10px 1px rgba(0, 0, 0, 0.14), 0px 2px 16px 1px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 4px 5px -2px rgba(0, 0, 0, 0.2), 0px 7px 10px 1px rgba(0, 0, 0, 0.14), 0px 2px 16px 1px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z8 {
  -webkit-box-shadow: 0px 5px 5px -3px rgba(0, 0, 0, 0.2), 0px 8px 10px 1px rgba(0, 0, 0, 0.14), 0px 3px 14px 2px rgba(0, 0, 0, 0.12);
  box-shadow: 0px 5px 5px -3px rgba(0, 0, 0, 0.2), 0px 8px 10px 1px rgba(0, 0, 0, 0.14), 0px 3px 14px 2px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z9 {
  -webkit-box-shadow: 0px 5px 6px -3px rgba(0, 0, 0, 0.2), 0px 9px 12px 1px rgba(0, 0, 0, 0.14), 0px 3px 16px 2px rgba(0, 0, 0, 0.12);
```

```
    box-shadow: 0px 5px 6px -3px rgba(0, 0, 0, 0.2), 0px 9px 12px 1px rgba(0, 0, 0, 0.14), 0px 3px 16px
2px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z10 {
    -webkit-box-shadow: 0px 6px 6px -3px rgba(0, 0, 0, 0.2), 0px 10px 14px 1px rgba(0, 0, 0, 0.14), 0px 4px
18px 3px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 6px 6px -3px rgba(0, 0, 0, 0.2), 0px 10px 14px 1px rgba(0, 0, 0, 0.14), 0px 4px 18px
3px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z11 {
    -webkit-box-shadow: 0px 6px 7px -4px rgba(0, 0, 0, 0.2), 0px 11px 15px 1px rgba(0, 0, 0, 0.14), 0px 4px
20px 3px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 6px 7px -4px rgba(0, 0, 0, 0.2), 0px 11px 15px 1px rgba(0, 0, 0, 0.14), 0px 4px 20px
3px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z12 {
    -webkit-box-shadow: 0px 7px 8px -4px rgba(0, 0, 0, 0.2), 0px 12px 17px 2px rgba(0, 0, 0, 0.14), 0px 5px
22px 4px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 7px 8px -4px rgba(0, 0, 0, 0.2), 0px 12px 17px 2px rgba(0, 0, 0, 0.14), 0px 5px 22px
4px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z13 {
    -webkit-box-shadow: 0px 7px 8px -4px rgba(0, 0, 0, 0.2), 0px 13px 19px 2px rgba(0, 0, 0, 0.14), 0px 5px
24px 4px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 7px 8px -4px rgba(0, 0, 0, 0.2), 0px 13px 19px 2px rgba(0, 0, 0, 0.14), 0px 5px 24px
4px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z14 {
    -webkit-box-shadow: 0px 7px 9px -4px rgba(0, 0, 0, 0.2), 0px 14px 21px 2px rgba(0, 0, 0, 0.14), 0px 5px
26px 4px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 7px 9px -4px rgba(0, 0, 0, 0.2), 0px 14px 21px 2px rgba(0, 0, 0, 0.14), 0px 5px 26px
4px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z15 {
```

```
-webkit-box-shadow: 0px 8px 9px -5px rgba(0, 0, 0, 0.2), 0px 15px 22px 2px rgba(0, 0, 0, 0.14), 0px 6px 28px 5px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 8px 9px -5px rgba(0, 0, 0, 0.2), 0px 15px 22px 2px rgba(0, 0, 0, 0.14), 0px 6px 28px 5px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z16 {
    -webkit-box-shadow: 0px 8px 10px -5px rgba(0, 0, 0, 0.2), 0px 16px 24px 2px rgba(0, 0, 0, 0.14), 0px 6px 30px 5px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 8px 10px -5px rgba(0, 0, 0, 0.2), 0px 16px 24px 2px rgba(0, 0, 0, 0.14), 0px 6px 30px 5px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z17 {
    -webkit-box-shadow: 0px 8px 11px -5px rgba(0, 0, 0, 0.2), 0px 17px 26px 2px rgba(0, 0, 0, 0.14), 0px 6px 32px 5px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 8px 11px -5px rgba(0, 0, 0, 0.2), 0px 17px 26px 2px rgba(0, 0, 0, 0.14), 0px 6px 32px 5px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z18 {
    -webkit-box-shadow: 0px 9px 11px -5px rgba(0, 0, 0, 0.2), 0px 18px 28px 2px rgba(0, 0, 0, 0.14), 0px 7px 34px 6px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 9px 11px -5px rgba(0, 0, 0, 0.2), 0px 18px 28px 2px rgba(0, 0, 0, 0.14), 0px 7px 34px 6px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z19 {
    -webkit-box-shadow: 0px 9px 12px -6px rgba(0, 0, 0, 0.2), 0px 19px 29px 2px rgba(0, 0, 0, 0.14), 0px 7px 36px 6px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 9px 12px -6px rgba(0, 0, 0, 0.2), 0px 19px 29px 2px rgba(0, 0, 0, 0.14), 0px 7px 36px 6px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z20 {
    -webkit-box-shadow: 0px 10px 13px -6px rgba(0, 0, 0, 0.2), 0px 20px 31px 3px rgba(0, 0, 0, 0.14), 0px 8px 38px 7px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 10px 13px -6px rgba(0, 0, 0, 0.2), 0px 20px 31px 3px rgba(0, 0, 0, 0.14), 0px 8px 38px 7px rgba(0, 0, 0, 0.12);
}

.mdc-elevation--z21 {
```

```

    -webkit-box-shadow: 0px 10px 13px -6px rgba(0, 0, 0, 0.2), 0px 21px 33px 3px rgba(0, 0, 0, 0.14), 0px 8px
40px 7px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 10px 13px -6px rgba(0, 0, 0, 0.2), 0px 21px 33px 3px rgba(0, 0, 0, 0.14), 0px 8px
40px 7px rgba(0, 0, 0, 0.12);
}
.mdc-elevation--z22 {
    -webkit-box-shadow: 0px 10px 14px -6px rgba(0, 0, 0, 0.2), 0px 22px 35px 3px rgba(0, 0, 0, 0.14), 0px 8px
42px 7px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 10px 14px -6px rgba(0, 0, 0, 0.2), 0px 22px 35px 3px rgba(0, 0, 0, 0.14), 0px 8px
42px 7px rgba(0, 0, 0, 0.12);
}
.mdc-elevation--z23 {
    -webkit-box-shadow: 0px 11px 14px -7px rgba(0, 0, 0, 0.2), 0px 23px 36px 3px rgba(0, 0, 0, 0.14), 0px 9px
44px 8px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 11px 14px -7px rgba(0, 0, 0, 0.2), 0px 23px 36px 3px rgba(0, 0, 0, 0.14), 0px 9px 44px
8px rgba(0, 0, 0, 0.12);
}
.mdc-elevation--z24 {
    -webkit-box-shadow: 0px 11px 15px -7px rgba(0, 0, 0, 0.2), 0px 24px 38px 3px rgba(0, 0, 0, 0.14), 0px 9px
46px 8px rgba(0, 0, 0, 0.12);
    box-shadow: 0px 11px 15px -7px rgba(0, 0, 0, 0.2), 0px 24px 38px 3px rgba(0, 0, 0, 0.14), 0px 9px 46px
8px rgba(0, 0, 0, 0.12);
}
nope {
    background-color: var(--whitebg);
    box-shadow: 1px 1px 3px #00000026;
    border-top: 1px solid var(--widgetbordercolor);
    border-bottom: 1px solid var(--widgetbordercolor);
    padding: 15px 0 15px 10px;
    overflow: hidden;
}
/* */
* {
    font-family: 'Overpass', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #1e1e21;
    color: #fff;
}
p{
    font-size: 14px;

```

```
}  
a {  
  text-decoration: none;  
  color: #fff;  
}  
a:hover {  
  text-decoration: none;  
  color: #fff;  
}  
.landing {  
  margin-top: 100px;  
}  
#navbarToggle {  
  background-color: #000;  
}  
.navbar.navbar-dark .nav-item .nav-link {  
  color: #fff!important;  
  background-color: #000;  
}  
.navbar.navbar-dark .nav-item .nav-link:hover {  
  color: #fff!important;  
  background-color: #000;  
}  
.navbar {  
  background-color: #000!important;  
}  
.navbar-nav {  
  background-color: #000;  
}  
.navbar-brand {  
  background-color: #000;  
}  
.lg {  
  background-color: #000;  
  font-size: xx-large;  
}  
h1 {  
  font-size: 33px;  
  color: #dfdcd9;
```

```
}  
h2 {  
  font-size: 1.5em;  
  color: #dfdcd9;  
}  
.author {  
  font-size: 12px;  
}  
.date {  
  font-size: 12px;  
}  
.media-scroller {  
  --_spacer: 3px;  
  display: grid;  
  gap: var(--_spacer);  
  grid-auto-flow: column;  
  grid-auto-columns: 21%;  
  
  padding: 0 var(--_spacer) var(--_spacer);  
  
  overflow-x: auto;  
  overscroll-behavior-inline: contain;  
}  
  .media-scroller--with-groups {  
    grid-auto-columns: 80%;  
  }  
  
.media-group {  
  display: grid;  
  gap: var(--_spacer);  
  grid-auto-flow: column;  
}  
  
.media-element {  
  display: grid;  
  grid-template-rows: min-content;  
  gap: var(--_spacer);  
  padding: var(--_spacer);  
  background: var(--surface-2);
```



```
border-radius: var(--radius-1);
box-shadow: var(--shadow-2);
}

.media-element > img {
  inline-size: 100%;
  aspect-ratio: 16 / 9;
  object-fit: cover;
}

.snaps-inline {
  scroll-snap-type: inline mandatory;
  scroll-padding-inline: var(--_spacer, 1rem);
}

.snaps-inline > * {
  scroll-snap-align: start;
}

.container {
  inline-size: min(100% - 4rem, 70rem);
  margin-inline: auto;
}

.flow {
  display: grid;
  gap: var(--size-3);
}

.page-header {
  padding-block: var(--size-9);
  margin-block-end: var(--size-9);
  background: var(--gradient-16);
  color: var(--gray-0);
  box-shadow: var(--shadow-2);
}

.page-title {
  font-size: var(--font-size-fluid-3);
}
```

```
.page-subtitle {
  font-size: var(--font-size-fluid-1);
}

.section-title {
  padding-inline-start: var(--size-6);
  margin-block: var(--size-9) var(--size-3);
}

.article-card {
  position: relative;
  display: -ms-flexbox;
  display: flex;
  -ms-flex-direction: column;
  flex-direction: column;
  min-width: 0;
  color: #000;
  background-color: #fff;
  word-wrap: break-word;
  background-clip: border-box;
  border-radius: 10px;
  margin-top: 15px;
}

.img {
  width: 100%;
}

.image-fluid {
  height: 18em;
}

.cardfooter {
  padding: 0.75rem 1.25rem;
  padding-bottom: 0px;
  border-radius: 0px 0px 10px 10px;
}

.article_name {
  color: #dfdcd9;
}

.source-body {
```

```
    margin-bottom: 30px;
}

.source {
    margin-top: 5px;
    padding-top: 5px;
    padding-bottom: 0px;
    margin-bottom: 0px;
    color: #dd0000;
}

.logo {
    color: #f3e742;
    background-color: #000;
}

.svg {
    background-color: #000;
}

.p-svg {
    background-color: #26262a;
}

.footer {
    background-color: #000;
    padding-top: 20px;
    padding-bottom: 15px;
}

.footer-text {
    background-color: #000;
    margin-bottom: 0px;
}

.dropdown-menu {
    background-color: #1e1e21;
}

.dropdown-item {
    color: #fff;
}

.dropdown-item:hover {
    color: #fff;
    background-color: #26262a;
}

.published {
```

```
background-color: #26262a;
}
.card-footer {
background-color: #26262a;
}
```

## Testing

### News\_test.py

```
import unittest
from ..models import Articles

class NewsTest(unittest.TestCase):
    def setUp(self):
        self.articles = Articles(
            "Techcrunch",
            "Steve Njuguna",
            "Nike acquires NFT collectibles studio RTFKT",
            "Nike is taking a plunge deeper into the world of crypto collectibles, announcing that they're acquiring the NFT studio RTFKT (pronounced 'artifact'). The acquisition announcement ...",
            "https://techcrunch.com/2021/12/13/nike-acquires-nft-collectibles-studio-rtfkt/",
            "https://techcrunch.com/wp-content/uploads/2021/12/FC3_itGXEEAA6z5g.jpg?w=1390&crop=1",
            "2021-12-13T20:19:00Z"
        )
    def test_instance(self):
```

```
        self.assertTrue(isinstance(self.articles, Articles))
if __name__ == "__main__":
    unittest.main()
```

## Source\_test.py

```
import unittest
from ..models import Sources

class SourceTest(unittest.TestCase):
    def setUp(self):
        self.source = Sources(
            "ABC News",
            "Your trusted source for breaking news, analysis, exclusive interviews, headlines, and videos at ABCNews.com.",
            "https://abcnews.go.com/"
        )
    def test_instance(self):
        self.assertTrue(isinstance(self.source, Sources))
if __name__ == "__main__":
    unittest.main()
```

## DockerFile

```
FROM ubuntu

RUN apt update
RUN apt install python3-pip -y
RUN pip3 install flask
RUN pip3 install ibm_db
RUN pip3 install flask_sqlalchemy
RUN pip3 install flask_bcrypt
RUN pip3 install flask_login
RUN pip3 install flask_wtf
RUN pip3 install wtforms
WORKDIR /main
COPY . .
CMD ["python3", "-m", "flask", "run", "--host=0.0.0.0"]
```



## **Source Code GitHub& Project Demo Link**

### **GitHub**

<https://github.com/IBM-EPBL/IBM-Project-39420-1660412540>

<https://github.com/IBM-EPBL/IBM-Project-39420-1660412540>

### **Project demo link**

[https://drive.google.com/file/d/1XWpCxpKXo4bj00WH5Pn3fE3Ef8Xihr3U/view?usp=share\\_link](https://drive.google.com/file/d/1XWpCxpKXo4bj00WH5Pn3fE3Ef8Xihr3U/view?usp=share_link)

[https://drive.google.com/file/d/1XWpCxpKXo4bj00WH5Pn3fE3Ef8Xihr3U/view?usp=share\\_link](https://drive.google.com/file/d/1XWpCxpKXo4bj00WH5Pn3fE3Ef8Xihr3U/view?usp=share_link)