

NEWS TRACKER APPLICATION

USING CLOUD

A Project Report submitted in Partial fulfilment of 7th Semester in degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Team ID : PNT2022TMID6775

ARTHI PA 67711931005

JANANI R 67711931019

KAMALESHWARI E 67711931023

SURYA KG 67711931050



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(NBA ACCREDITED)

GOVERNMENT COLLEGE OF ENGINEERING ,SALEM- 11

ANNA UNIVERSITY: CHENNAI 600025

NOV-2022

GOVERNMENT COLLEGE OF ENGINEERING ,SALEM 11
(AN AUTONOMOUS INSTITUTION AFFILIATED TO ANNA UNIVERSITY,CHENNAI*NAAC ACCREDITED)



BONAFIDE CERTIFICATE

Certified that this project report “**NEWS TRACKER APPLICATION**” is the Bonafide Record work done by **ARTHI PA (67711931005), JANANI R (67711931019), KAMALESHWARI E(67711931023), SURYA KG(67711931050)**, for **IBM NALAYATHIRAN** in **VII Semester of B.E.,** degree course in **Electronics and Communication Engineering** Branch during the Academic year of 2022-2023.

Staff -In charge

Prof. M. PRIADARSINI M.E.,

Evaluator

Prof K.JAYANTHI M.E.,

HEAD OF THE DEPARTMENT

Dr .A.M. KALPANA M.E, PHD ..,

ACKNOWLEDGEMENT

We express our breathless thanks to our **Dr.MALAYALAMOORTHY M.E.,Phd..**, the PRINCIPAL, GOVERNMENT COLLEGE OF ENGINEERING SALEM for giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department **Dr.A.M KALPANA M.E,Phd..**, for giving us valuable suggestion and help towards us throughout this project

We are highly grateful to thank our Project coordinator **PROF K.NITHYA M.E..**, and our Project Evaluator **K.JAYANTHI M.E..**, Department of Electronics and Communication Engineering, Government college of Engineering, Salem for the coordinating us throughout this Project.

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

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

ARTHI P A

JANANI R

KAMALESHWARI E

SURYA K G

ABSTRACT

Everyone has the right to freedom of speech. However, this right is being misused to differentiate and attack others, physically or verbally, in the name of free speech. This discrimination is known as hate speech. Hate speech can be well-defined as language used to express hate towards a person or a group of people based on characteristics such as race, religion, ethnicity, gender, nationality, disability and sexual orientation. The increasing usage of social sites and information sharing has specified major benefits to humanity. However, this has also assumed rise to a variety of challenges including the spreading and sharing of hate speech messages. Thus, to solve this emerging issue in social media sites, recent studies employed a variety of machine learning and deep learning algorithms with text mining algorithm to automatically detect the hate speech messages on real time datasets. Hence, the aim of this Project is to analyses the comments on social networks using Natural Language processing technique (NLP) and Deep learning algorithm named as Back propagation neural network algorithm. Using NLP technique, can extract the keywords from user generated content and implement Back Propagation neural network to classify the text whether it is positive or negative. If it is negative means, automatically block the comments as per user wish and also block the friends based on pre-defined threshold values. Experimental results shows that the proposed framework implemented in real time social network site with improved notification.

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

5. PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database 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

1.INTRODUCTION

Mobile app ecosystems are transforming patterns of news consumption. Until quite recently, reading the news was a niche use for smartphones, 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; a similar picture is found in the U K. This growth in mobile news access continues the migration of news consumers to the Internet.

1.1 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.

1.2Purpose:

Newspapers are one of the most popular and most needed commodities in our daily life. In today's busy world, reading newspapers has become one of the traditional ways of getting news. News is produced every minute and distributed via television, radio and the Internet, so the news updated the next morning is already outdated. So, newspaper and magazine publishers have a hard time keeping up with the pace. Change is needed and publishers must embrace mobile.

2.LITERATURE SURVEY

2.1References:

Like articles, websites, blogs. <https://www.researchgate.net/>.

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

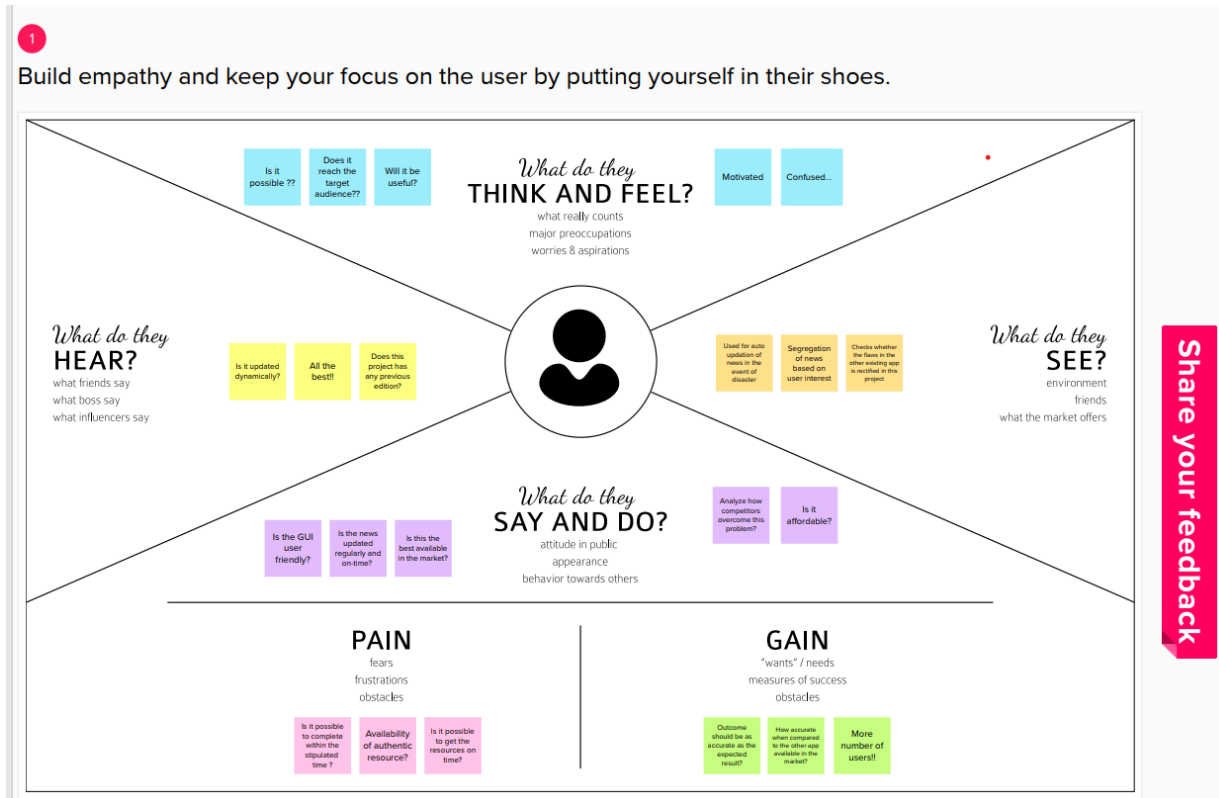
<https://www.ripublication.com/>

2.2 Problem Statement Definition:

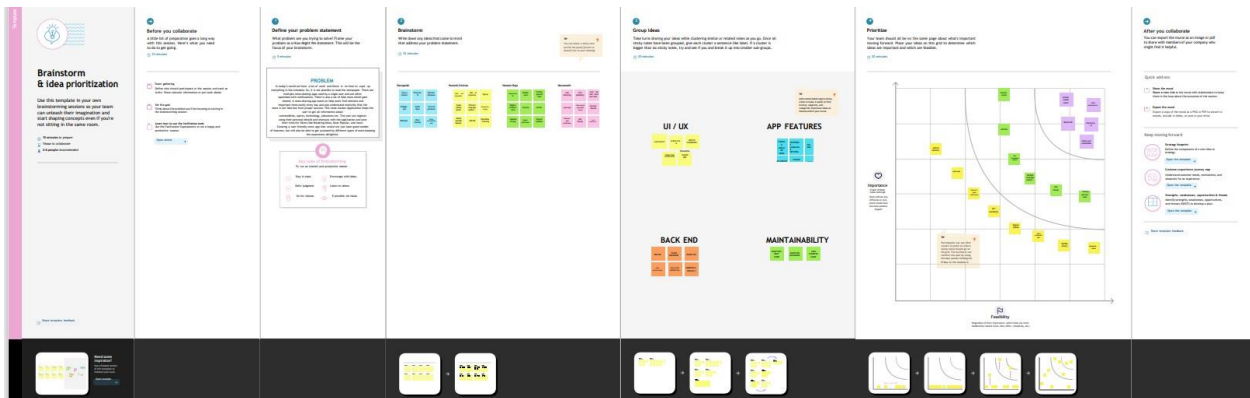
In today's world we have a lot of work and there is no time to cope up everything in the schedule. So, it is not possible to read the newspaper. By using this application, the user can access the news they are interested and get a quick peek for the day. There are multiple news-sharing apps used by a single user and are often spammed with notifications. There is also a lot of fake news which gets shared. A news sharing app wants to help users find relevant and important news easily every day and also understand explicitly that the news is not fake but from proper sources. This news tracker application helps the user to get all information about commodities, sports, technology, education etc. The user can register using their personal details and interacts with the application and save their time. • Creating an app that benefits with the smart news filtering functionality where it allows its users to segregate various news sections, comprising of business, sports, political, international, and more. The users can opt for filters like Breaking News, Most Popular, and more.

3.IDEATION & PROPOSED SOLUTION:

3.1. Empathy Map Canvas:



3.2. Ideation & Brainstorming



3.3. Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	The user needs a way to get relevant news based on his choices so that the user does not have to spend a lot of time on searching news. News is filled with ads and spams annoys and irritates the user and affects the user experience.
2.	Idea solution/Description	A cloud-based web application which enables the user to get the news as per their interests, choices and location. News will be filtered as per the user's wish and it will be displayed to the user
3.	Novelty/Uniqueness	This news tracker has features that enables the user to view news about crimes and health updates happening in and around their location. This will be given the most priority as it is important to every person to be aware of recent happenings. The tracker app does not have annoying ads.

4.	Social Impact/Customer satisfaction	The user will not have to spend a lot of time in searching for the relevant news. The customer will be finally able to view news without ads Since we are using the News API, the news will be reliable and accurate. The customer can be aware of recent happenings.
5.	Business model/ Revenue model	The model will generate revenue through offering few premium features to user. The features include seeing stories from the user's favourite publishers, regional news, few regional languages. The default language will be English.
6.	Scalability of the solution	Since the web application is deployed on IBM cloud, it can handle multiple users at a time. The user will go through a seamless experience and it enables them to view the news according to their interests and choices. Users from all age category can use the application and the news can also be filtered according to their age

3.4 PROBLEM SOLUTION FIT:

PROBLEM STATEMENT

- A trader who needs daily updates of cryptocurrencies and stock market to trade and invest ,our news application helps him to give detailed updated headlines daily as well as instantly .
- A Civil service exam aspirant who needs the headlines of international affairs and domestic affairs ,our app solves this issue by giving him the news which he require.
- A fishermen who requires daily weather updates , our news application helps them by providing latest update on weather so that they can make plan of visiting sea for fishing and reach shore safely.
- A young enthusiast who is furious Neymor fan he wants to find daily updates of football world cup our app gets this details of the matches as well as the updates of every match.
- A traveller who wants to know the updated news of every place which he wants to visit instantly about the rates, weather, new places, our apps give him the updated news of all the things which he require to travel happily.
- A techie guy who likes to know daily updates on new launches of computers and other electronics , so our application help him to get daily launches of different devices from different brands to full fill his needs.

4.REQUIREMENT ANALYSIS

4.1Functional requirement

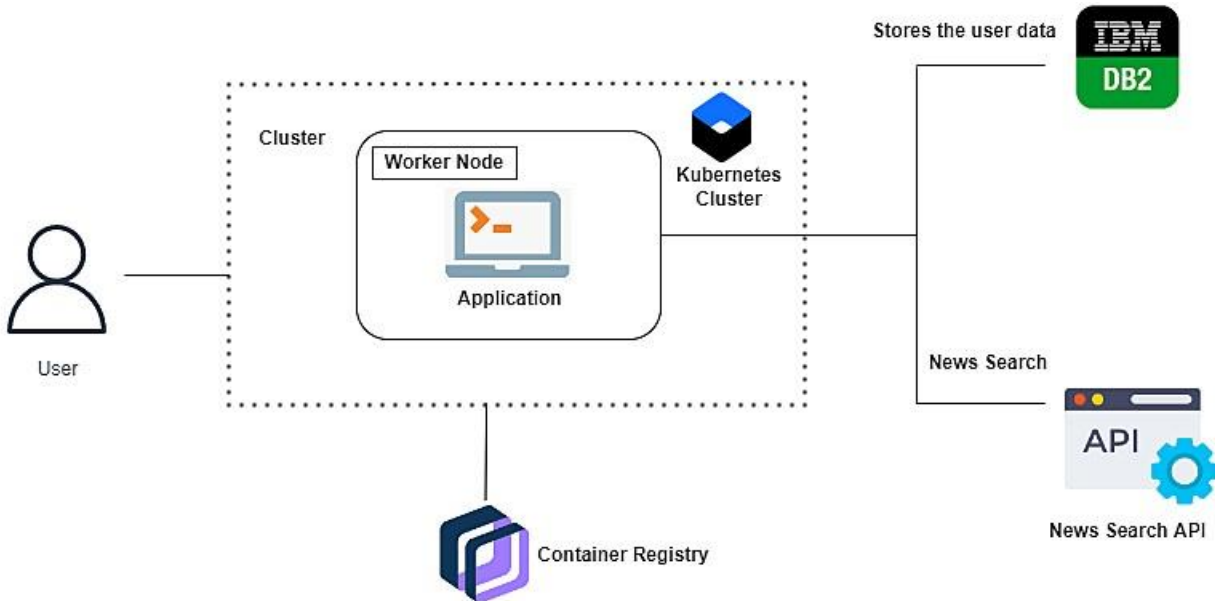
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Installation	User can install the app from Google play store or from the website
FR-2	User Registration	Registration through Form Registration through Gmail
FR-3	User Confirmation	Confirmation via Email
FR-4	User Login	User should login the app with the user's name and password

4.2Non-Functional requirements

FR No	Non-Functional Requirement	Description
NFR-1	Usability	Everyone can understand the process of using the app easily by the commands given in the app
NFR-2	Security	It is a more secured app. No fake news can be shared.
NFR-3	Performance	Performance of the app is very great
NFR-4	Availability	More sub categories are available

5.PROJECT DESIGN

5.1Data Flow Diagrams

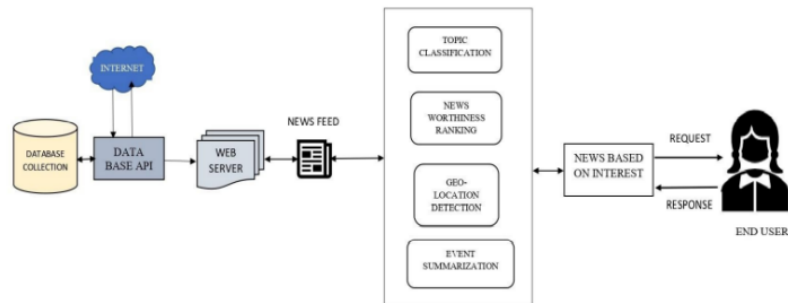


5.2. Solution & Technical Architecture

Solution architecture is a complex process – with many subprocesses – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements to the project.
- Provide specifications according to which the solution is defined, managed, and delivered to the customers.

Example - Solution Architecture Diagram:



NEWS TRACKER APPLICATION

5.3. User Stories:

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.

6.PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation:

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

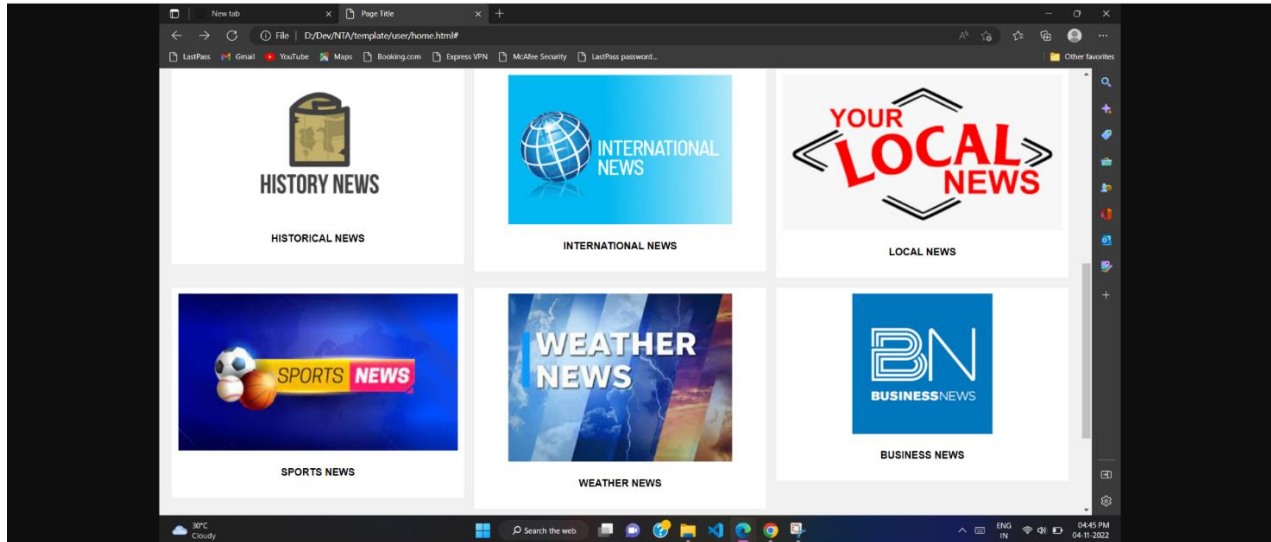
Spr.n:	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Janani, Kamaleshwari
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Surya,Arthi
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	Kamaleshwari, Surya
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Arthi,Janani
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Janani, Kamaleshwari
Sprint-2	Dashboard	USN-6	As a user I should be able to navigate and access all the features hassle free	2	High	Surya, Arthi
Sprint-2	Layout	USN-7	As a user I should be able to access the portal with different devices with the same comfort	2	High	Janani,surya
Sprint-3	Data Store and retrieval	USN-8	Get Data from API and store as JSON in DB2	3	High	Arthi, Kamaleshwari
Sprint-3		USN-9	Get bin data from API and store in DFS	2	High	Janani,Arthi

6.2. Sprint Delivery Schedule

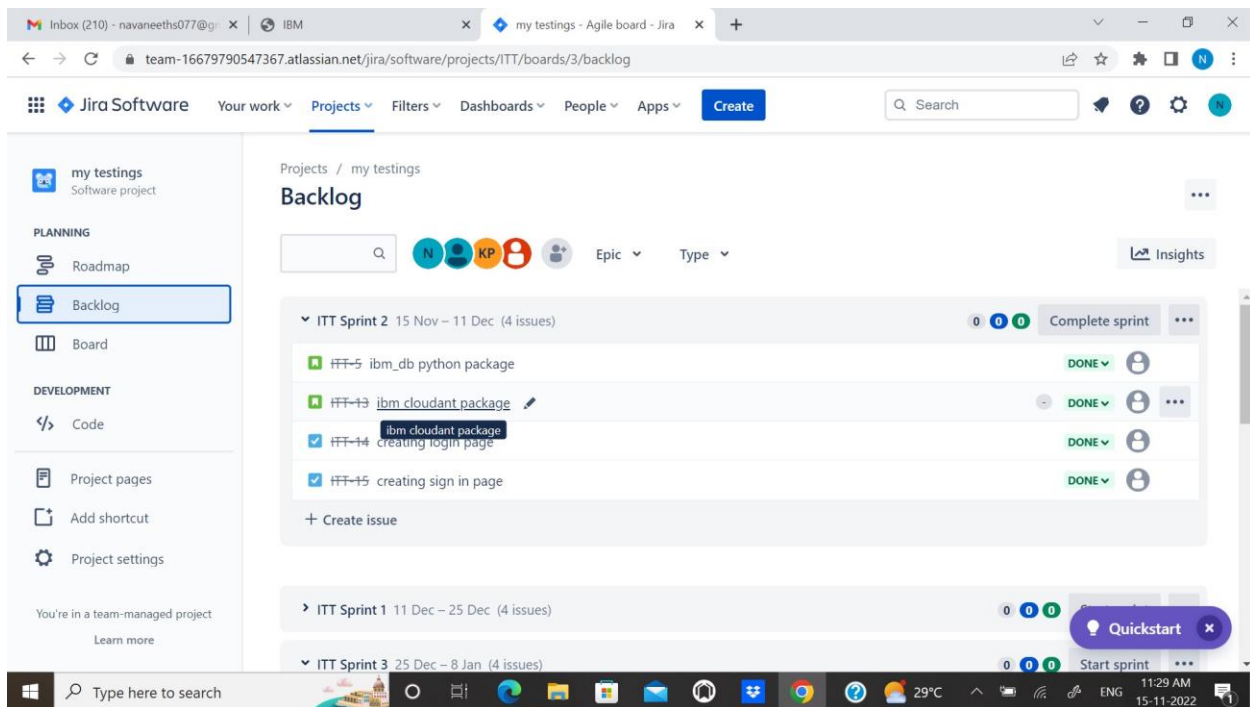
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	9 Nov 2022	11 Nov 2022	20	11 Nov 2022
Sprint-2	20	6 Days	11 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-3	20	6 Days	13 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	20	6 Days	15 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3. Reports from JIRA

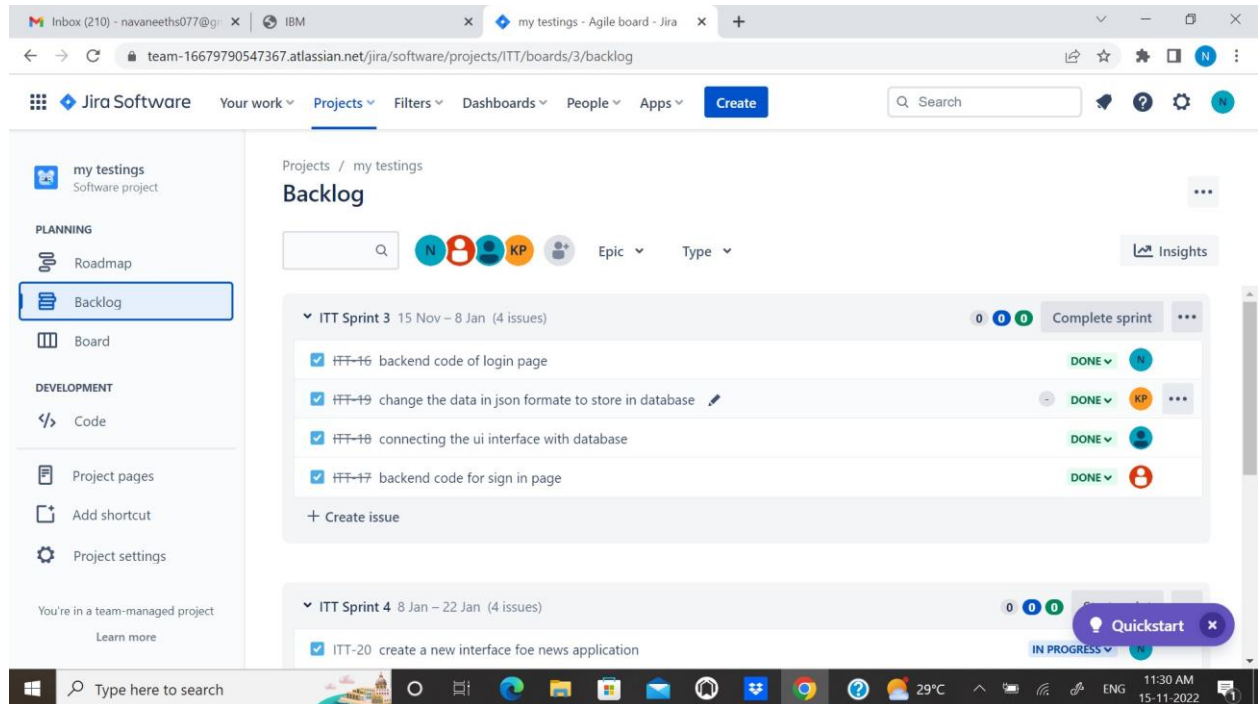
Sprint-1



Sprint-2



Sprint-3



The screenshot shows the Jira Software interface for the 'my testings' project. The left sidebar contains navigation options: Roadmap, Backlog (selected), Board, Code, Project pages, Add shortcut, and Project settings. The main area displays the 'Backlog' for 'ITT Sprint 3' (15 Nov - 8 Jan, 4 issues). The issues are:

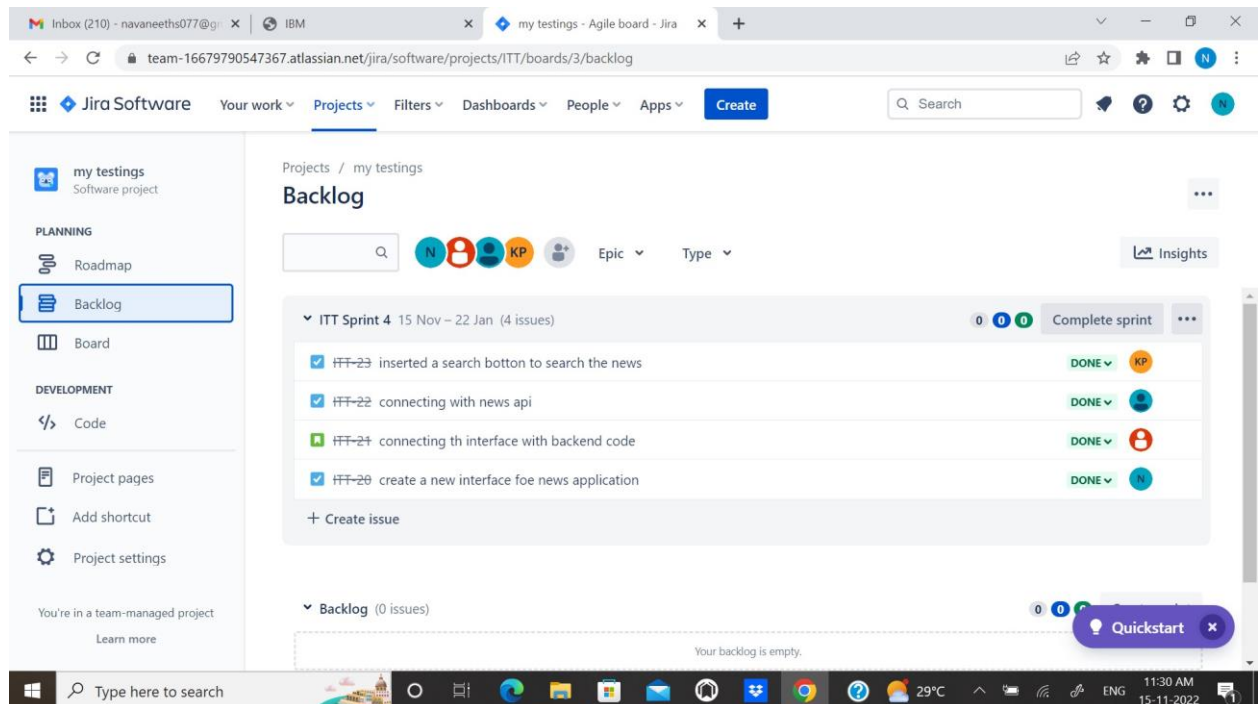
- IT-16: backend code of login page (DONE)
- IT-19: change the data in json format to store in database (DONE)
- IT-18: connecting the ui interface with database (DONE)
- IT-17: backend code for sign in page (DONE)

Below the sprint, there is a section for 'ITT Sprint 4' (8 Jan - 22 Jan, 4 issues) with one issue:

- IT-20: create a new interface for news application (IN PROGRESS)

A 'Quickstart' button is visible in the bottom right corner of the sprint section.

Sprint-4



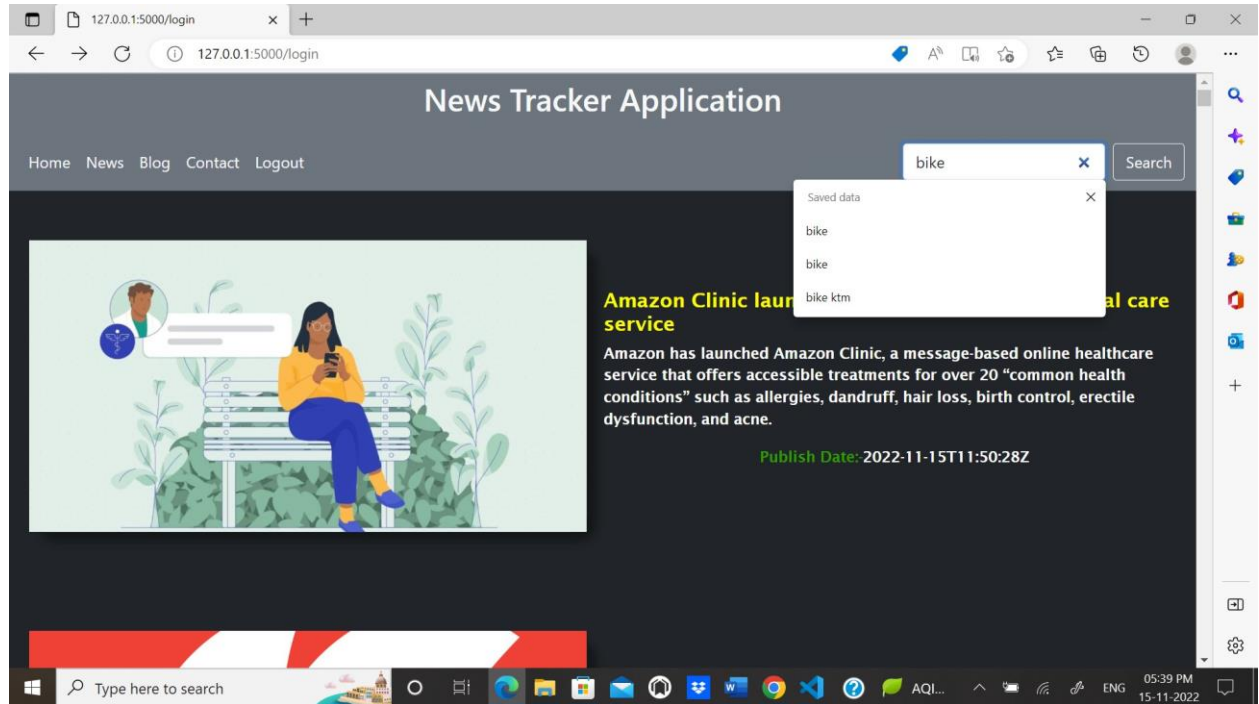
The screenshot shows the Jira Software interface for the 'my testings' project. The left sidebar contains navigation options: Roadmap, Backlog (selected), Board, Code, Project pages, Add shortcut, and Project settings. The main area displays the 'Backlog' for 'ITT Sprint 4' (15 Nov - 22 Jan, 4 issues). The issues are:

- IT-23: inserted a search button to search the news (DONE)
- IT-22: connecting with news api (DONE)
- IT-24: connecting th interface with backend code (DONE)
- IT-26: create a new interface for news application (DONE)

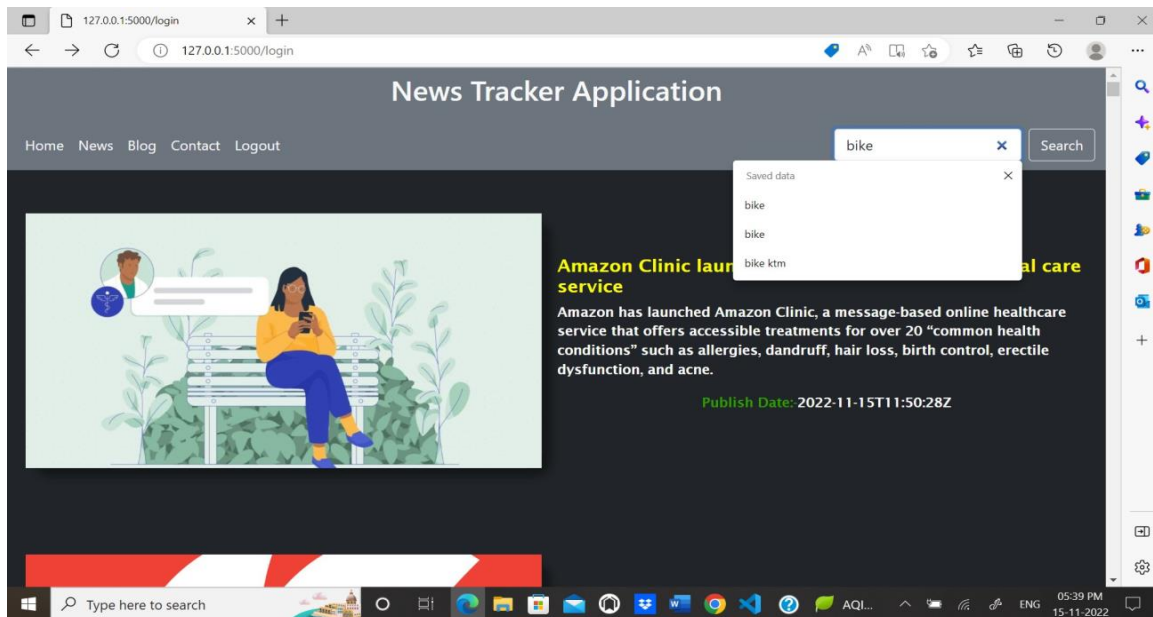
Below the sprint, there is a section for 'Backlog' (0 issues) with the message 'Your backlog is empty.' A 'Quickstart' button is visible in the bottom right corner of the sprint section.

7.CODING & SOLUTIONING

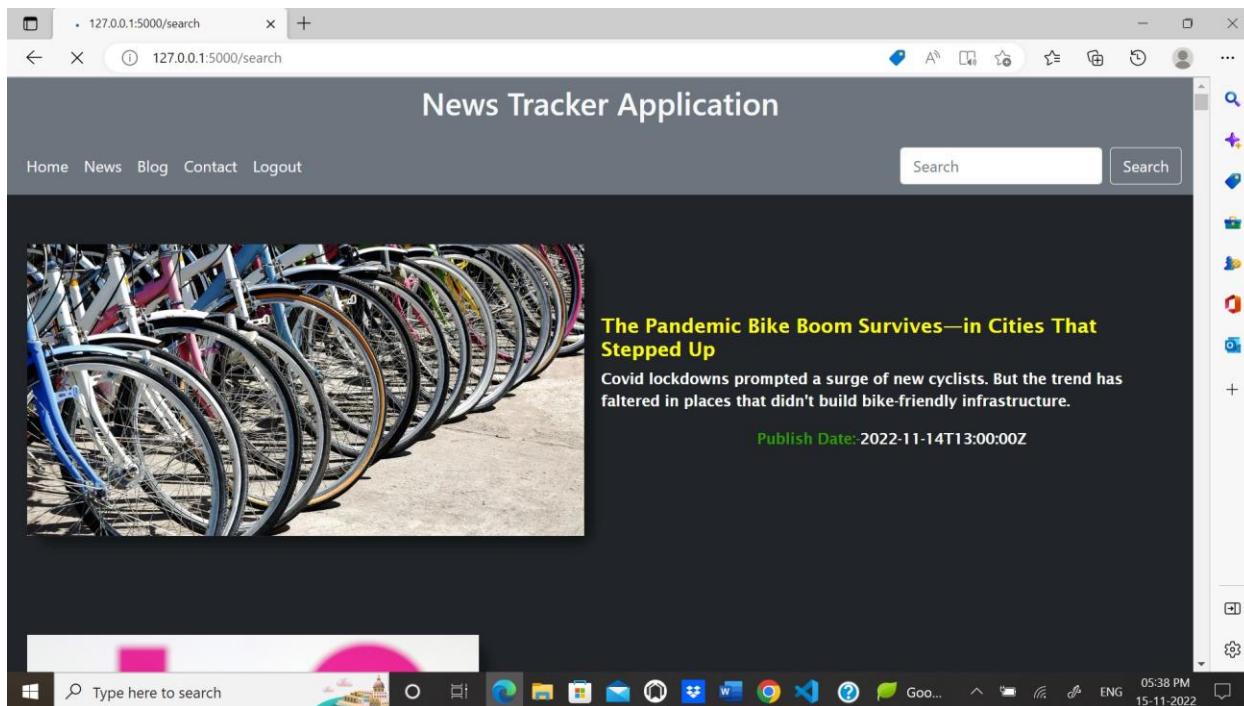
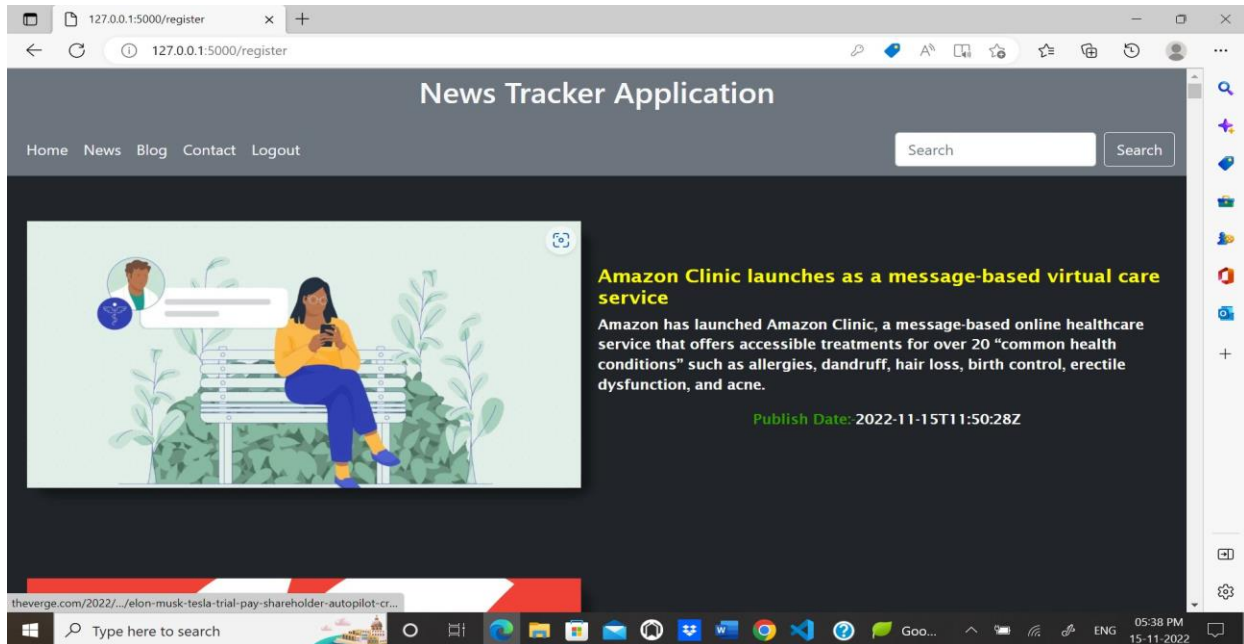
7.1Feature 1:



7.2 Feature 2:



7.3 Database Schema:



8.TESTING

8.1. User Test case:

User Test case

DATE	18 NOV 2022
TEAM ID	PNT2022TMID6775
PROJECT	NEWS TRACKER APPLICATION

Test case	Feature	Components	Test scenario	Expected result	Actual result	Status	Comment	Bug
Sign in	Functional	Login page	Verify user can see sign in option	Can visible	Yes visible	Pass	Successful	-
Sign up	Functional	Login page	Verify user can sign up	Can visible	Yes visible	Pass	Successful	-
Forgot news	Functional	Login page	Verify user can see forgot password	Option is available	Option is available	Pass	Successful	-
Fetch news	Functional	Home page	Verify user can see the news	News will be feed to the app	404 error	Fail	Un Successful	-
Type of news available in the page	Functional	Fetch news	Verify types of news in the app	Economy, sports, weather	Hover buttons will be shown.	pass	Successful	-

8.2 User Acceptance Testing:



Date	18 NOV 2022
Team ID	PNT2022TMID6775
project	NEWS TRACKER APPLICATION



1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

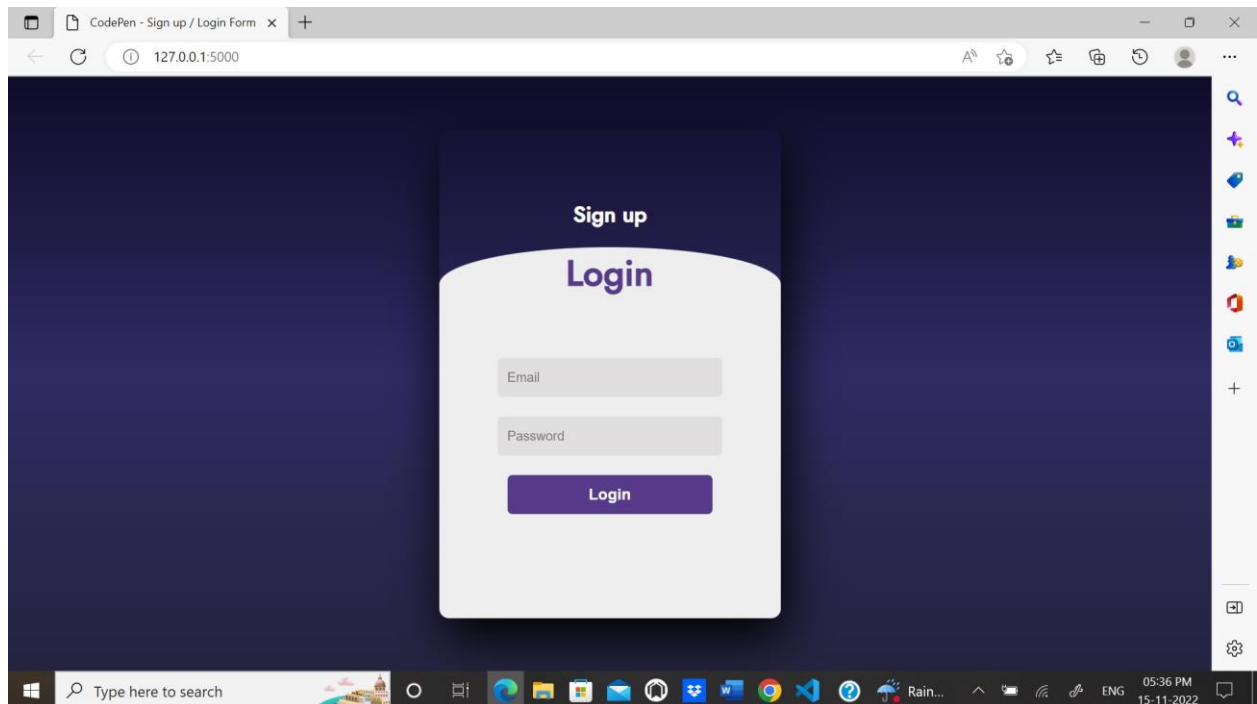
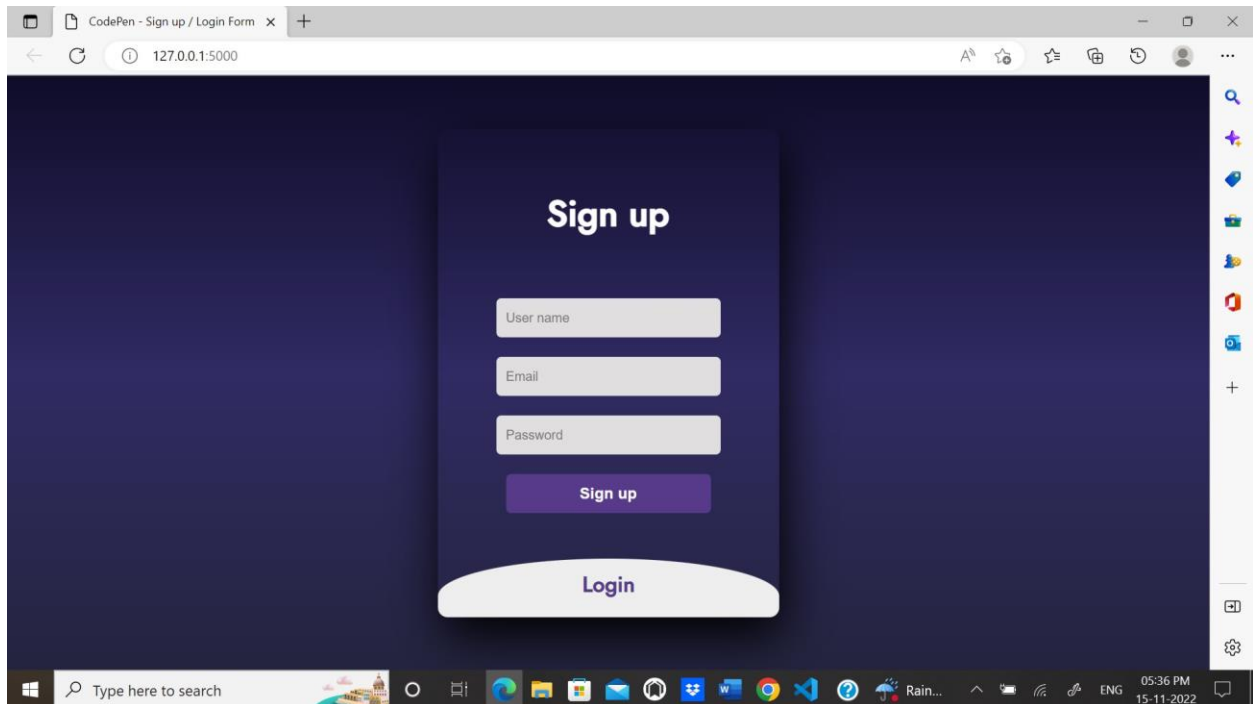
3. Test Case Analysis

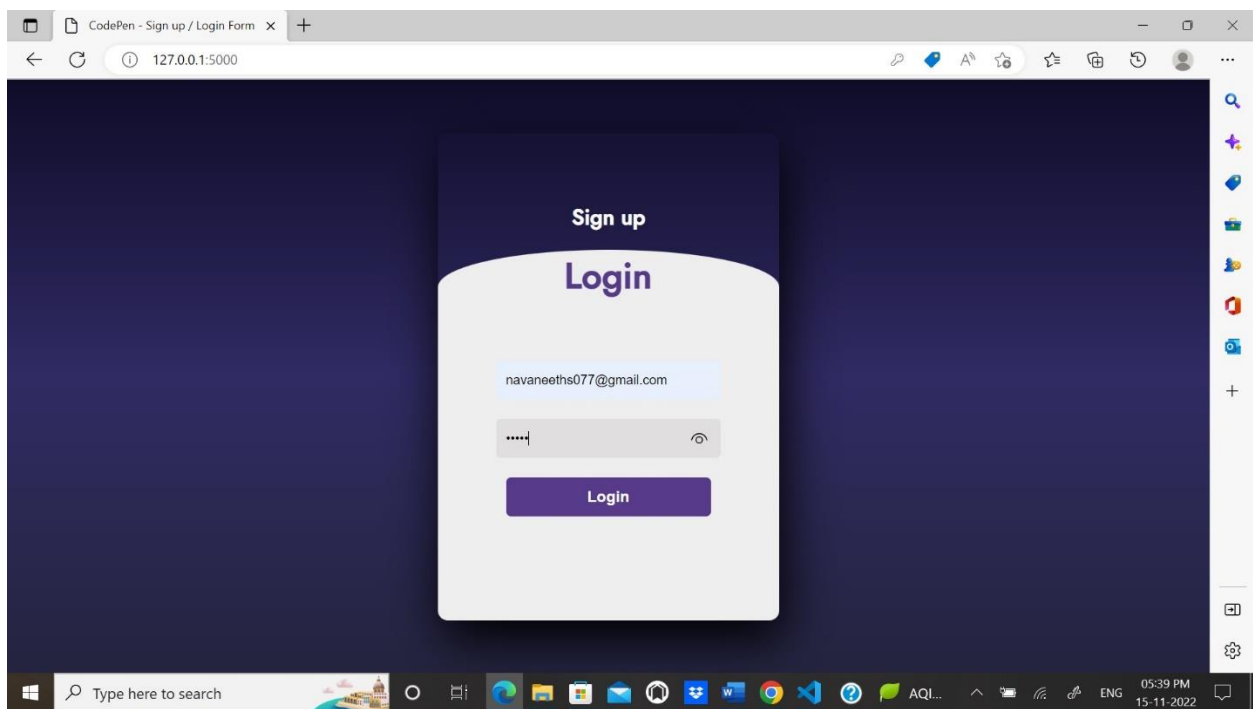
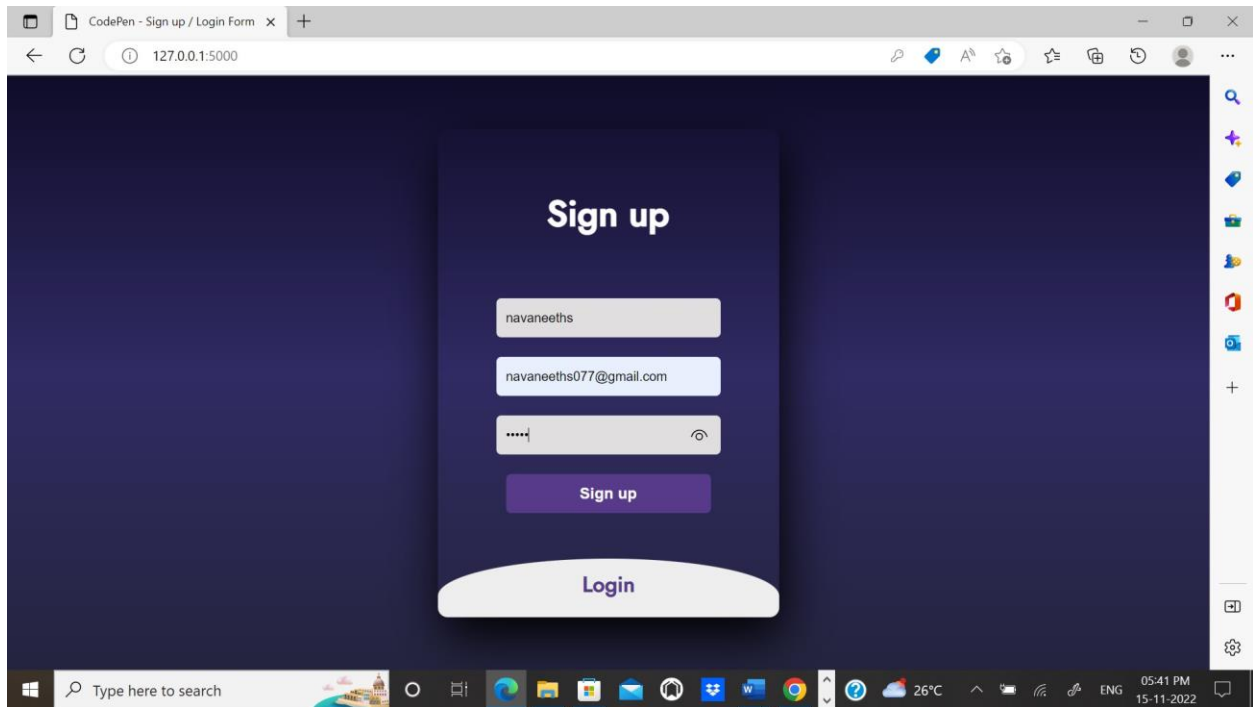
This report shows the number of test cases that have passed, failed, and untested

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

9.RESULTS

9.1Performance Metrics:





10.ADVANTAGES

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

10.DISADVANTAGES

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

11.CONCLUSION

Thus, the projected using IBM CLOUD is tested, verified and executed successfully.

12.Future scope

In future, we planned to implement this project in large scale which will be helpful and used by all the people.

13.APPENDIX

13.1. Source Code

Home.html

```
<!DOCTYPE html>
```

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

```
<head>
```

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

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <link rel="stylesheet" href="{{url_for('static', filename='css/style.css')}}">
```

```
  <link rel="preconnect" href="https://fonts.googleapis.com">
```

```
  <link rel="icon" href="{{url_for('static', filename='img/logo.png')}}">
```

```
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

```
<title>{% block title %}{% end block %}</title>
```

```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
```

EVSTQN3/azprG1Anm3QDgpJlIm9Nao0Yz1ztcQTWfSpd3yD65VohhpUuCOmLASjC"
crossorigin="anonymous">

</head> <body>

```
<nav class="bg-secondary">
  <h2 class="logo text-light">News Tracker Application</h2>
  <!-- <ul class="menu">
    <li class="nav-items"><a href="/">Home</a></li>
    <li class="nav-items"><a href="/">News</a></li>
    <li class="nav-items"><a href="/">Blogs</a></li>
    <li class="nav-items"><a href="/">Contact</a></li>
  </ul> -->
</nav>
```

```
<nav class="navbar navbar-expand-lg bg-secondary">
  <div class="container-fluid">
    <!-- <h2>News Tracker Application</h2> -->
    <!-- <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bstarget="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
arialabel="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button> -->
    <div class="collapse navbar-collapse" id="navbarSupportedContent">

      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active text-light" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-light" href="#">News</a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-light" href="#">Blog</a>
        </li>

        <li class="nav-item" role="contact">
          <a class="nav-link text-light" href="/contact">Contact</a>
        </li>
```



```

        <li class="nav-item" rol="logout" >
            <a class="nav-link text-light" href="/">Logout</a>
        </li>
    </ul>

    <form class="d-flex" role="search" action = "/search" method = "post">
        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search" name =
"nm">
        <button class="btn btn-outline-light" type="submit">Search</button>
    </form>
</div>
</div>
</nav>
<section id="main" class="bg-dark">
    <!-- <form action = "/search" method = "post">
        <p><input type = "text" name = "nm" /></p>
        <p><input type = "submit" value = "submit" /></p>
    </form> -->

                                <!--                                <div                                class="banner">
                                </div> -->
    <!-- <div class = "head">TODAY'S TOP HEADLINES</div>
    {% for title,desc,img,p_date,url in contents %}
    <div class="wrapper">
        <div class="item"><a href="{{url}}"></a></div>        <div
class = "c-box">
        <div class="content"><a href="{{url}}"><h2>{{title}}</h2></a></div>
        <div class="content"><a href="{{url}}"><p>{{desc}}</p></a></div>
                                <div class="content"><p
style="color:rgb(77,    255,    0);">Publish    Date:- </p><p>{{p_date}}</p></div>
        </div>
    </div>
    {% endfor %} -->
    {% for title,desc,img,p_date,url in all %}
    <div class="wrapper">
        <div class="item"><a href="{{url}}"></a></div>        <div
class = "c-box">
        <div class="content"><a href="{{url}}"><h2>{{title}}</h2></a></div>
        <div class="content"><p>{{desc}}</p></div>
        <div class="content"><p style="color:rgb(48, 158, 0);">Publish Date:- </p><p> {{p_date}} </p></div>
    </div>

```

```

    </div>
{% endfor %}
</section>

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-
IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-cVKIPhGWIC2Al4u+LWgxfKTRlcfu0JTxR+EQDz/bglDoEyl4H0zUF0QKbrJ0EcQF"
crossorigin="anonymous"></script>
</body>
</html>

```

Index.html

```

<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>CodePen - Sign up / Login Form</title>
  <link rel="stylesheet" href="{{url_for('static', filename='css/style2.css')}}">

</head>
<body>
<!-- partial:index.partial.html -->
<!DOCTYPE html>
<html>
<head>
  <title>Slide Navbar</title>
  <link rel="stylesheet" href="{{url_for('static', filename='css/style2.css')}}">
<link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap" rel="stylesheet">
</head>
<body>
  <div class="main">
    <input type="checkbox" id="chk" aria-hidden="true">

    <div class="signup">
      <form action = "/register" method="post">
        <label for="chk" aria-hidden="true">Sign up</label>
        <input type="text" name="un" placeholder="User name"
required="">

        <input      type="email"      name="ue"      placeholder="Email"
required="">

```

```

        <input type="password" name="up" placeholder="Password"
required="">

        <button type="submit">Sign up</button>
    </form> </div>

    <div class="login">
        <form action = "/login" method="post">
            <label for="chk" aria-hidden="true">Login</label>
            <input type="email" name="email" placeholder="Email"
required="">

            <input type="password" name="pswd" placeholder="Password"
required="">

            <button type="submit">Login</button>
        </form>
    </div>
</div>
</body>
</html>
<!-- partial -->
</body>
</html>

```

Contact.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>Document</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpUuCOmLASjC"
crossorigin="anonymous">

</head>
<body>
    <div class='h-100 d-flex align-items-center justify-content-center'>
        <table class="table table-bordered table-dark">
            <tr>

                <th scope="col">Team Member</th>

```

	Contact (Click the Symbol)
Naveen Raja N	
Navaneeth S	

```

</tr>
<tr>
<td>Kousick Srinivas P</td>
<td>

<a href="https://www.linkedin.com/in/kousick-srinivas-b978a11ab"><svg
xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-linkedin"
viewBox="0 0 16 16">
<path d="M0 1.146C0 .513.526 0 1.175 0h13.65C15.474 0 16 .513 16 1.146v13.708c0 .633-.
.526 1.146-1.175 1.146H1.175C.526 16 0 15.487 0 14.854V1.146zm4.943
12.248V6.169H2.542v7.225h2.401zm-1.2-8.212c.837 0 1.358-.554 1.358-1.248-.015-.709-.52-1.248-
1.342-1.248-.822 0-1.359.54-1.359 1.248 0 .694.521 1.248 1.327 1.248h.016zm4.908 8.212V9.359c0.216.016-
.432.08-.586.173-.431.568-.878 1.232-.878.869 0 1.216.662 1.216 1.634v3.865h2.401V9.25c0.22-1.184-
3.252-2.764-3.252-1.274 0-1.845.7-2.165 1.193v.025h-.016a5.54 5.54 0 0 1
.016.025V6.169h-2.4c.03.678 0 7.225 0 7.225h2.4z"/>
</svg>
</a>
</td>
</tr>
<tr>
<td>Karuppiyah AL </td>
<td>

<a href="https://www.linkedin.com/in/karuppiyah-al-97b405184"><svg
xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-linkedin"
viewBox="0 0 16 16">
<path d="M0 1.146C0 .513.526 0 1.175 0h13.65C15.474 0 16 .513 16 1.146v13.708c0 .633-.
.526 1.146-1.175 1.146H1.175C.526 16 0 15.487 0 14.854V1.146zm4.943
12.248V6.169H2.542v7.225h2.401zm-1.2-8.212c.837 0 1.358-.554 1.358-1.248-.015-.709-.52-1.248-
1.342-1.248-.822 0-1.359.54-1.359 1.248 0 .694.521 1.248 1.327 1.248h.016zm4.908 8.212V9.359c0.216.016-
.432.08-.586.173-.431.568-.878 1.232-.878.869 0 1.216.662 1.216 1.634v3.865h2.401V9.25c0.22-1.184-
3.252-2.764-3.252-1.274 0-1.845.7-2.165 1.193v.025h-.016a5.54 5.54 0 0 1
.016.025V6.169h-2.4c.03.678 0 7.225 0 7.225h2.4z"/>
</svg>
</a>
</td>
</tr>
</table>
</div>

```

```

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-
IQsoLXI5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-cVKIPhGWIC2Al4u+LWgxfKTRlcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
crossorigin="anonymous"></script>
</body>
</html>

```

Style.css body

```

{
    margin: 0px;
    padding: 0px; background-color:
    rgb(69, 54, 54);
}
* {
    box-sizing: border-box;
}
ul {
    list-style: none;
} a
{
    text-decoration: none! important;
} nav
{
    min-height: 60px;
    width: 100%;
    /* background-color: red; */
    /* border-top: 1px solid black! important; */
    position: relative;

    display: flex;
    justify-content: space-around;
    align-items: center;
    flex-flow: row wrap;
}
.logo h2 {
    font-size: 30px;
    font-weight: bold;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell,
'Open Sans', 'Helvetica Neue', sans-serif;
}

```

```

    color: white !important;
    text-decoration: none !important;
    letter-spacing: 1px;
}

```

```

nav .menu {
    display: flex;
    flex-flow: row wrap;
    justify-content: center;
    align-items: center;
} nav .menu li a
{ height:
35px;
    line-height: 30px;
    display: flex ;
    text-transform: uppercase;
    align-items: center;
    padding: 0px 22px;
    font-size: 15px;
    font-weight: bold; color:
    white; font-family:
    'Poppins', sans-serif;
    letter-spacing: 1px;
}
nav .menu li a:hover {
    color: black;
    background-color: white;
    border-radius: 20px;
}
#main {
    /* padding-top: 98px; */
}
.banner {
    height: 620px;
    width: 100%;
    display: flex;
    flex-wrap: column;
    justify-content: center;
    align-items: center;
    background-color: black;
    border: 1px solid rgb(98, 4, 4);
}

```

```

}
.banner img {
  height: 100%;
  width: 100%;
  padding: 2px;
} .head-git{
  height: 40px;
  width: 100%;
  justify-content: center;
  align-items: center;
  background-color: rgb(250, 244, 244);
  display: flex;
  flex-flow: row;
  margin-bottom: 5px;
}
.head-git h2 {
  font-size: 18px;
  font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
  color:black;
}
.head-git h2 a {
  text-decoration: none ;
  color:blue;
  font-size: 23px; }
.head-git h2 a:hover { color:rgb(211,
  47, 41); background-color: beige;
}
.nav-link {
  height: 100%;
  width: 100%;
  color: black;
}
.head {
  height: 50px;
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-size: 30px;
  font-weight: bold;

```



```

    font-family:'Times New Roman', Times, serif;
    padding-left: 20px;
    padding-top: 10px;
}
.wrapper {
    height: 400px;
    width: 100%;
    display: flex;
    align-items: center;
    padding: 10px;
    justify-content: space-between;
    /* background-color: rgb(68, 47, 47); */
}
.wrapper .item {
    height: 300px;
    min-width: 30%;
    border-radius: 20px;
    margin: 10px;
    box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.4);
    padding: 0.5px;
    flex-flow: column wrap;
    border-bottom: 1px white; }
.wrapper .item img {
    height: 100%;
    width: 100%; }
.c-box {
    height: content;
    max-width: 50%;
    display: flex; align-items:
    center; padding: 10px;
    flex-flow: column;
    padding: 1px;
}
.c-box .content {
    height: content;
    min-width: 30%;
    display: flex;
    align-items: center;
    flex-flow: row wrap;
}
.c-box .content p {

```

```

color: #fff;
font-size: 15px;
font-weight: bold;
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif; }
.c-box .content h2 {
color: yellow;
font-size: 20px;
font-weight: bold;
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif;
} .head-f{
height: 40px;
width: 100%;
justify-content: center;
align-items: center;
border-top: 2px solid black;
background-color: yellow;
display: flex;
flex-flow: column;
}
.head-f h2 {
font-size: 15px;
font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
color: black;
}

```

```

Style2.css body{ margin: 0;
padding: 0; display: flex;
justify-content: center; align-
items: center; min-height:
100vh; font-family: 'Jost',
sans-serif; background: linear-
gradient(to bottom, #0f0c29,
#302b63, #24243e);
}

```

```

.main{ width: 350px;
height: 500px;
background: red;
overflow: hidden;
background:

```

[url\("https://doc-08-2c-](https://doc-08-2c-)

docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dmis5/fo0picsp1nhiucmc
 0l2
 5s29respgpr4j/1631524275000/03522360960922298374/03522360960922298374/1Sx0jhdpEpnNlydS4
 r
 nN4kHSJtU1EyWka?e=view&authuser=0&nonce=gcrocepgbb17m&user=03522360960922298374&has
 h=tfhgbs86ka6divo3llbvp93mg4csvb38") no-repeat center/ cover; border-radius: 10px;
 box-shadow: 5px 20px 50px #000;
 } #chk{ display: none;
 } .signup{
 position: relative;
 width:100%;
 height: 100%;
 } label{ color: #fff; font-size:
 2.3em; justify-content:
 center; display: flex; margin:
 60px; font-weight: bold;
 cursor: pointer;
 transition: .5s ease-in-out;
 } input{ width: 60%; height:
 20px; background: #e0dede;
 justify-content: center;
 display: flex; margin: 20px
 auto; padding: 10px; border:
 none; outline: none; border-
 radius: 5px;
 } button{ width:
 60%;
 height: 40px; margin:
 10px auto; justify-
 content: center;
 display: block; color:
 #fff; background:
 #573b8a; font-size:
 1em; font-weight:
 bold; margin-top:
 20px; outline: none;
 border: none; border-
 radius: 5px;
 transition: .2s ease-in;
 cursor: pointer;
 } button:hover{ background:
 #6d44b8;

```

} .login{ height: 460px; background:
#eee; border-radius: 60% / 10%;
transform: translateY(-180px);
transition: .8s ease-in-out;
}
.login label{ color: #573b8a;
transform:
scale(.6);
}

#chk:checked ~ .login{ transform:
translateY(-500px);
}
#chk:checked ~ .login label{ transform:
scale(1);
}
#chk:checked ~ .signup label{ transform:
scale(.6);
} app.py
from cloudant.client import Cloudant from
cloudant.error import CloudantException from
cloudant.result import Result, ResultByKey from
flask import Flask, render_template, request
from newsapi import NewsApiClient

serviceUsername = "50ab9a31-993d-4eaf-abf1-21c6f737b08c-bluemix" servicePassword =
"6864bc1c41f5701310846a0a80700140c138e7c82b8c7f678ce5ab877f8b90b9" serviceURL
= "https://50ab9a31-993d-4eaf-abf1-21c6f737b08c-
bluemix:6864bc1c41f5701310846a0a80700140c138e7c82b8c7f678ce5ab877f8b90b9@50ab9a31-993d4eaf-
abf1-21c6f737b08c-bluemix.cloudantnosqldb.appdomain.cloud"

client = Cloudant(serviceUsername, servicePassword, url=serviceURL) client.connect()

app = Flask(__name__)

@app.route("/news")
def news():
    api_key = '4f625485702d4929a546bcf4eb9d5c79'

    newsapi = NewsApiClient(api_key=api_key)

```

```
top_headlines = newsapi.get_top_headlines(sources = "the-verge")
all_articles = newsapi.get_everything(sources = "the-verge")
```

```
t_articles = top_headlines['articles']
a_articles = all_articles['articles']
```

```
news = []
desc = []
img = []
p_date = []
url = []
```

```
for i in range(len(t_articles)):
    main_article = t_articles[i]
```

```
    news.append(main_article['title'])
    desc.append(main_article['description'])
    img.append(main_article['urlToImage'])
    p_date.append(main_article['publishedAt'])
    url.append(main_article['url'])
```

```
    contents = zip( news,desc,img,p_date,url)
```

```
news_all = []
desc_all = []
img_all = []
p_date_all = []
url_all = []
```

```
for j in range(len(a_articles)):
    main_all_articles = a_articles[j]
```

```
    news_all.append(main_all_articles['title'])
    desc_all.append(main_all_articles['description'])
    img_all.append(main_all_articles['urlToImage'])
    p_date_all.append(main_all_articles['publishedAt'])
    url_all.append(main_article['url'])
```

```
    all = zip( news_all,desc_all,img_all,p_date_all,url_all)
```

```
return render_template('home.html',all = all)
```

```

@app.route("/search",methods = ['POST', 'GET']) def
searchFunc():
    inputText = request.form['nm']
    api_key = '4f625485702d4929a546bcf4eb9d5c79'

    newsapi = NewsApiClient(api_key=api_key)

    top_headlines = newsapi.get_top_headlines(sources="bbc-news")
    all_articles = newsapi.get_everything(q=inputText)

    t_articles = top_headlines['articles']
    a_articles = all_articles['articles']

    news = []
    desc = []
    img = []
    p_date = []
    url = []

    for i in range (len(t_articles)):
        main_article = t_articles[i]

        news.append(main_article['title'])
        desc.append(main_article['description'])
        img.append(main_article['urlToImage'])
        p_date.append(main_article['publishedAt'])
        url.append(main_article['url'])

        contents = zip( news,desc,img,p_date,url)

    news_all = []
    desc_all = []
    img_all = []
    p_date_all = []
    url_all = []

    for j in range(len(a_articles)):
        main_all_articles = a_articles[j]
        news_all.append(main_all_articles['title'])
        desc_all.append(main_all_articles['description'])
        img_all.append(main_all_articles['urlToImage'])

```

```

    p_date_all.append(main_all_articles['publishedAt'])
    url_all.append(main_article['url'])

    all = zip( news_all,desc_all,img_all,p_date_all,url_all)

    return render_template('home.html', all = all)

def addNewUser(userName,userEmail,userPassword):
    jsontdata = {}
    jsontdata["userName"] = str(userName)
    jsontdata["userEmail"] = str(userEmail)
    jsontdata["userPassword"] = str(userPassword)

    myDataBase = client['database1']
    newDocument = myDataBase.create_document(jsontdata)

def authenticate(userName,userEmail):
    myDataBase = client['database1']
    result_collection = Result(myDataBase.all_docs, include_docs=True)
    for data in result_collection:

        if data['doc']['userName'] == str(userName):
            return True
        if data['doc']['userEmail'] == str(userEmail):
            return True
    return False

def authenticateLogin(userEmail,userPassword):
    myDataBase = client['database1']
    result_collection = Result(myDataBase.all_docs, include_docs=True)

    for data in result_collection:
        if data['doc']['userPassword'] == str(userPassword) and data['doc']['userEmail'] == str(userEmail):
            return True
    return False

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

    userEmail = request.form.get("email")

```

```

userPassword = request.form.get("pswd")

if(authenticateLogin(userEmail,userPassword)):
    return news()
return render_template("index.html")

@app.route("/register",methods = ['POST', 'GET']) def
registerUserData():
    userName = request.form.get("un")
    userEmail = request.form.get("ue")
    userPassword = request.form.get("up")

    print(userEmail,userName,userPassword)
    if(authenticate(userName=userName,userEmail=userEmail)):
        return render_template("index.html")
    addNewUser(userName,userEmail,userPassword)
    return news()

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/contact")
def hh():
    return render_template("contact.html")

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

```

GitHub:

<https://github.com/IBM-EPBL/IBM-Project-23749-1659924902>

Project Demo Link: https://drive.google.com/file/d/1in0rKi_zmliEqf2i7FbzWht8fNHRXSyJ/view?usp=drivesdk