

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 : PNT2022TMID43908

PRIYANKA V	721919106060
KARTHIKEYAN P	721919106039
ARUN R	721919106014
GOBIANANDH A	721919106029

INDUSTRY MENTOR : SAI PRIYA

FACULTY MENTOR : MUKUNTHAN S

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. News Tracker is a full stack web application which allows users to register along with their favourite topics, upon login the app displays the news based on the user's interest. The news displayed in the app is based on the Newcatcher API and Cricbuzz API from Rapid API site. A news-sharing app wants to help users find relevant and important news easily every day and also provide explicitly news from that users locality/region which may of help to the user.

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. Enabling users to view news from anywhere at anytime. It also helps to reduce the time to get information about a specific topic. Also enables a person to get an updated news which may help Business people to make business related decisions quickly and correctly

2.LITERATURE SURVEY :

News is one of the primary source of gaining information about the actions and events that happen all around. It may be an event that happened in the past, happening now or going to happen in the future. In the present days where there is a rapid increase in the development and adaptability of technologies throughout all the demographic of people, it is necessary to provide news in such a way that it is interconnected with the current technological trends. 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, here, we will take a look at all the previous solutions, attempts and implementations to the news tracker application or anything that is atleast vaguely related to it.

2.1 EXISTING PROBLEM

NewsBreak is a popular website to read ongoing and past news via the internet browsers. The

website works by aggregating news from various sources and presents them in a likeable

manner for the users to read it.

The website also offers the ability for users to sign up to the so said website and record

their progress, manage profiles ,bookmark commenting on news ends and so on.

2.2References: Like articles, websites, blogs. <https://www.researchgate.net/>.

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

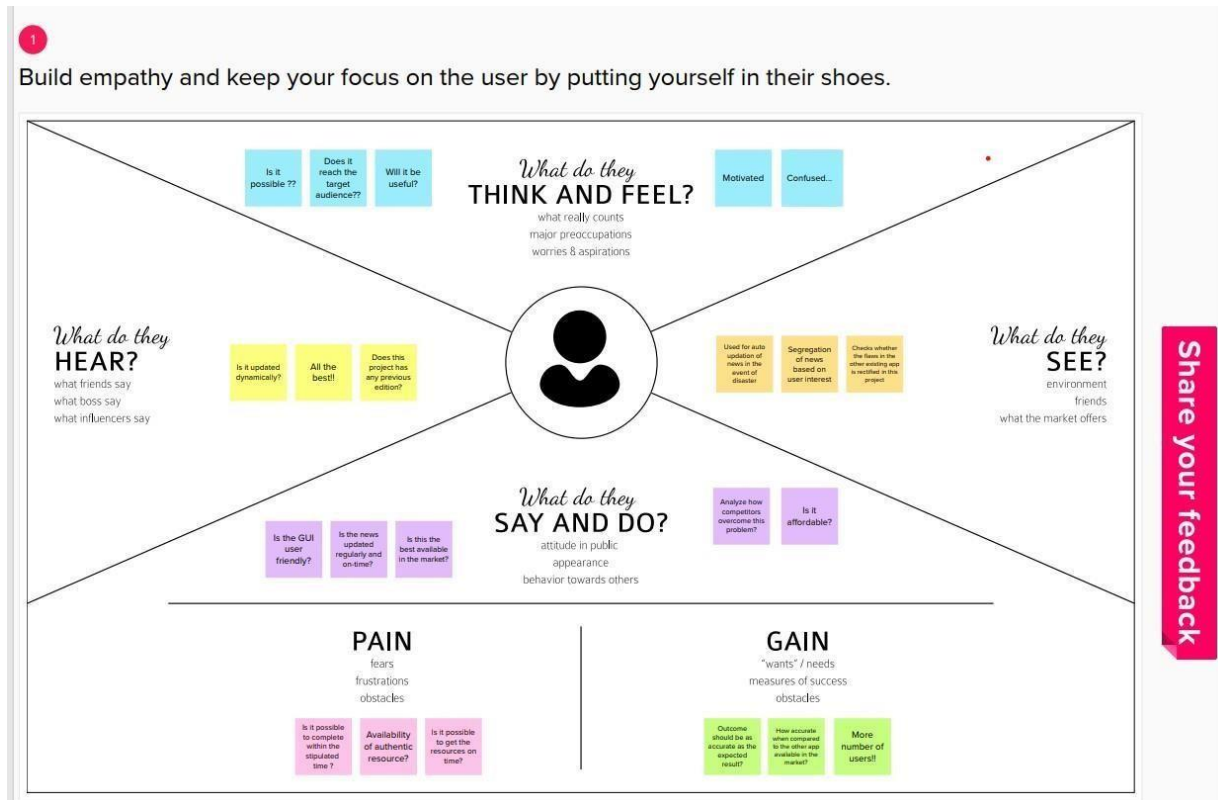
<https://www.ripublication.com/>

2.3Problem 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 newssharing 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:

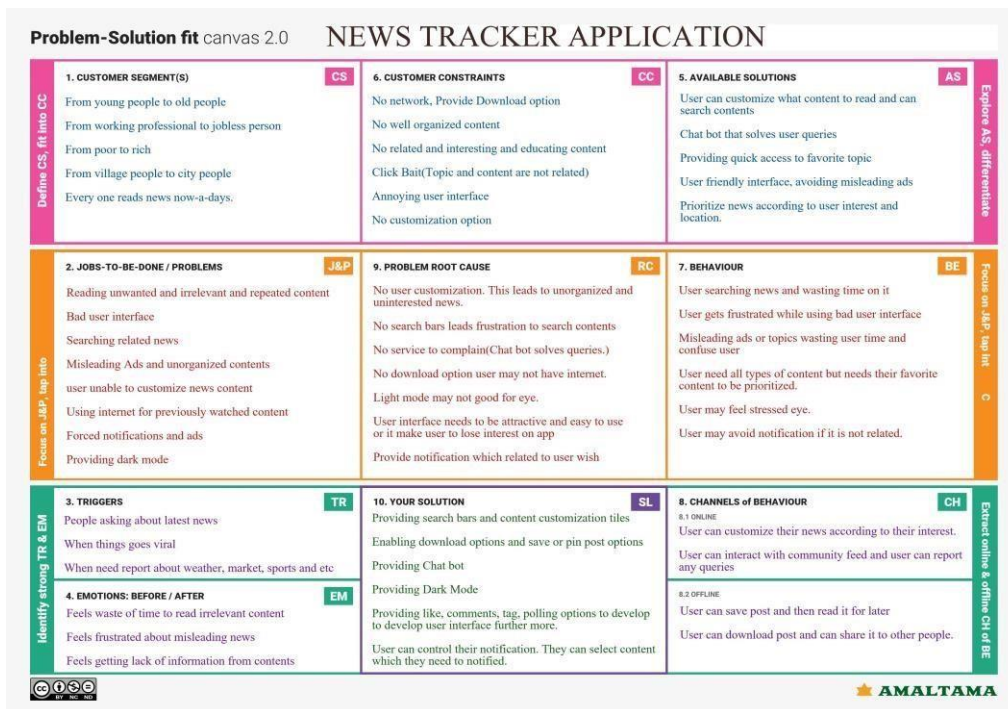


3.3. Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	Most people don't like to carry a newspaper with them. Some people want them to be updated only in the area they are interested in.
2.	Idea solution/Description	An application needs to be developed in which users can read news whenever they want and they will be able to customize their area of interest. So that they will be notified, if any new news is updated in their interested areas.
3.	Novelty/Uniqueness	<ol style="list-style-type: none">1. A user can read news only from their interested fields rather than reading all the news.2. This application provides users with a trusted and secured ecosystem. News shared through the application is original and spam free.
4.	Social Impact/Customer satisfaction	This application encourages its users to provide feedback. Based on that feedback, developments were made eventually.

5.	Business model/ Revenue model	<p>1. Add advertisements to the application, so that we can get revenue from those advertisement-sponsored organizations. More advertisements may irritate the user.</p> <p>2. Add premium subscription, users who subscribe for premium won't get advertisements.</p>
6.	Scalability of the solution	As it was an application-based project, correct ideation and execution can develop an application with no bugs and errors, so that the user might like our application and some might suggest and share it to their surroundings, resulting in an increase in our application insights.

3.4 PROBLEM SOLUTION FIT:



4.REQUIREMENT ANALYSIS

4.1Functional 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 Gmail
FR-2	User confirmation	Confirmation via Google Auth
FR-3	User login	User login with username and password Login with Gmail
FR-4	User interface	User login with username and password Login with Gmail

4.2 Non-Functional requirements

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

FR No	Non-Functional Requirement	Description
NFR-1	Usability	The application is easily navigable, it has an interface which is user-friendly. The login/register page has simple terms which can be filled by a user who has not done it before. The user interface navigates the user to the Home page and asks for interests. The settings can be quickly accessed by the user by going to the account settings, where the credentials and interests can be changed anytime.
NFR-2	Security	The application uses Google OAuth 2.0 / Auth0 for User authentication.
NFR-3	Reliability	The application should be able to perform without failure in 95 percent of all use cases during a month since the IBM cloud doesn't face down time so often
NFR-4	Availability	The application must be available to the users at least 99% of the time in a month.

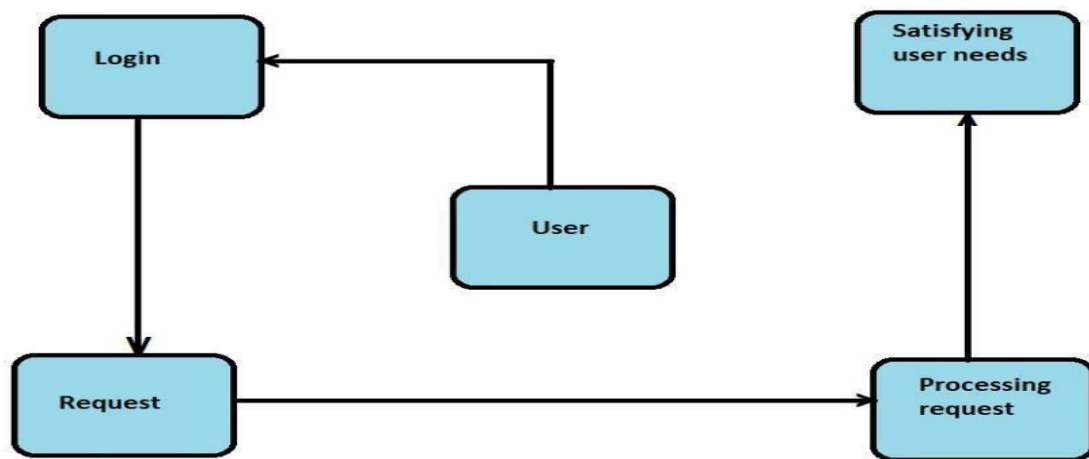
5.PROJECT DESIGN

5.1Data Flow Diagrams

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

Simplified:

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



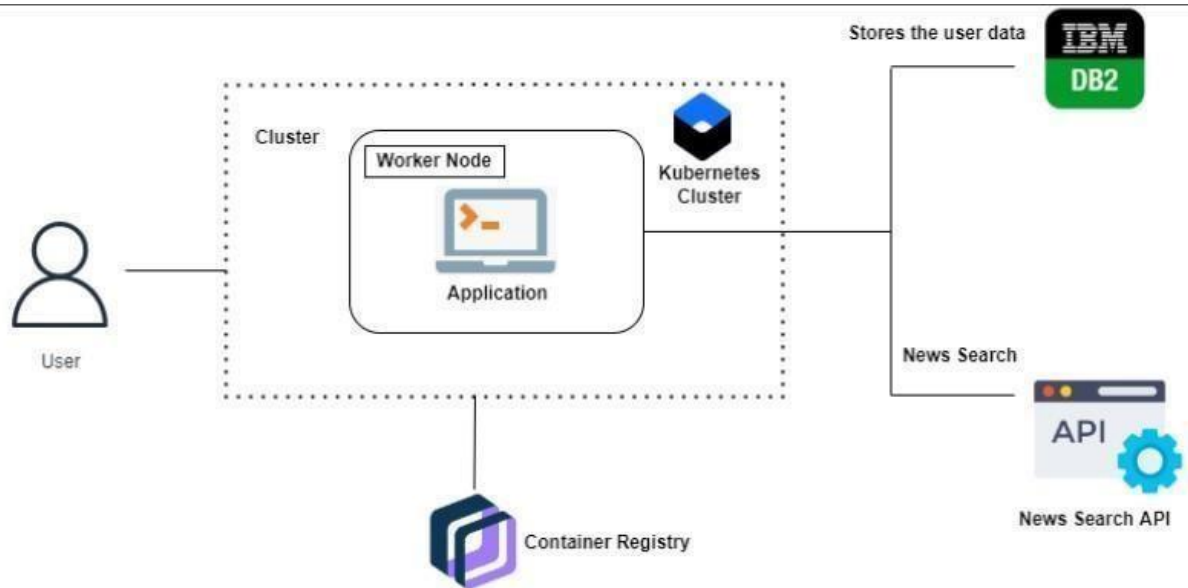
		USN-3	As a user, I can register for the application through Gmail	I can register through Gmail by OTP Authentication	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can view all types of information through this application	High	Sprint-1
	Dashboard	USN-5	To see their histories about recently viewed, updates for search related news, current progress, feedback		Medium	Sprint-2
Customer (Web user)	Browser	USN-6	Works as an interactive medium between client and server	I can access the resources through browser	High	Sprint-1
Customer Care Executive	Chat bot	USN-7	Rectify the customer's issues related to account, subscription and customization	Chat bot can resolve simple issues for customers	low	Sprint-2

Feedback	Feedback Form	USN-8	Getting feedback from customers helps application's administrator to improve the quality of the application	Customers can tell their Opinions	High	Sprint-1
Administrator	Admin module	USN-9	As an admin, I will modify the application as per customer requirements and fix the bugs to give customers a bug free service	I can modify the entire application	High	Sprint-2

5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub processes – 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.



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:

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
Sprint-2	Confirmation Email	USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
Sprint-3	User profile	USN-3	Once the registration done,a separate profile will be created for a user and they can access the information securely from their profile.	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
Sprint-4	Search the information	USN-4	After the profile creation as a user,I can search the news information to be needed.	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
Sprint-5	Category	USN-5	A user can search the information by selecting their category like sports,food,politics,weather etc....	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-6	Location	USN-6	As per user location or selected location by user,the information will be feeded.	1	Medium	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH

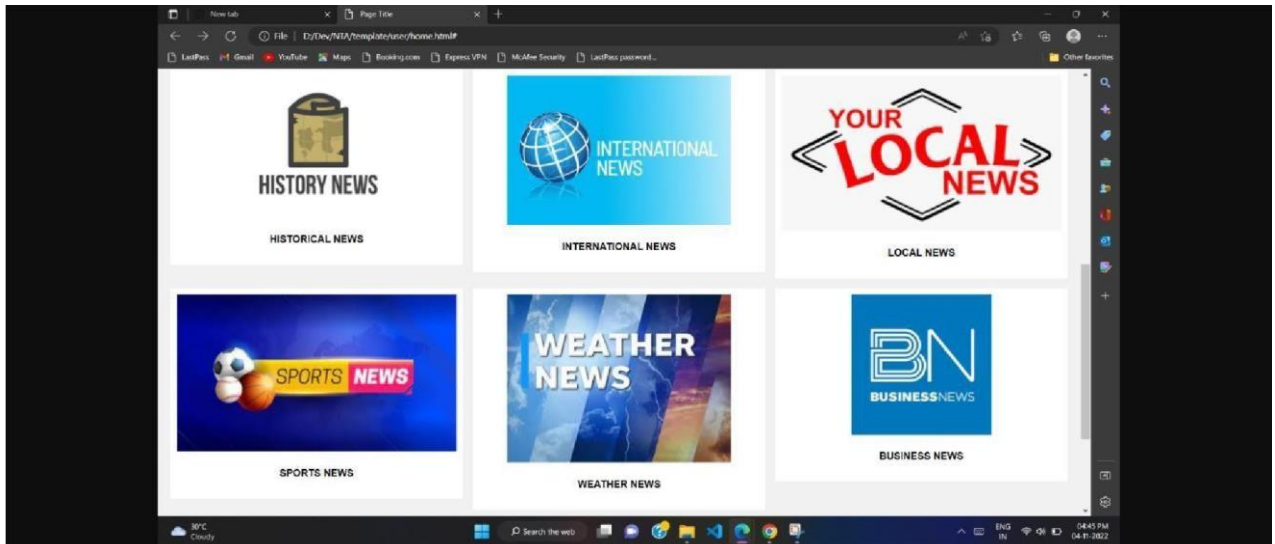
Sprint-7	Language	USN-7	As a user,I can read or view the information as my needed language.	2	High	V. PRIYANKA P. KARTHIKEYAN R. ARUN A. GOBIANANDH
----------	----------	-------	---	---	------	---

6.2. Sprint Delivery Schedule

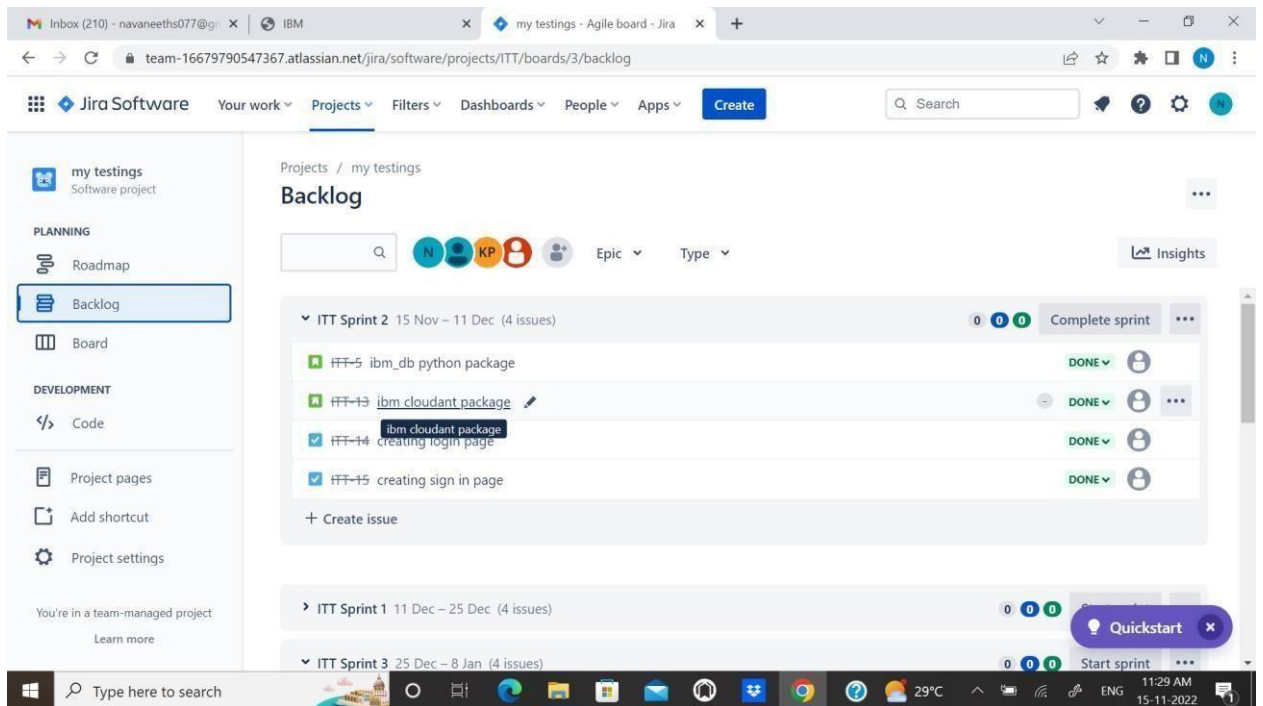
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24Oct 2022	29 Oct 2022	20	29Oct 2022
Sprint-2	20	6 Days	31Oct2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3. Reports from JIRA

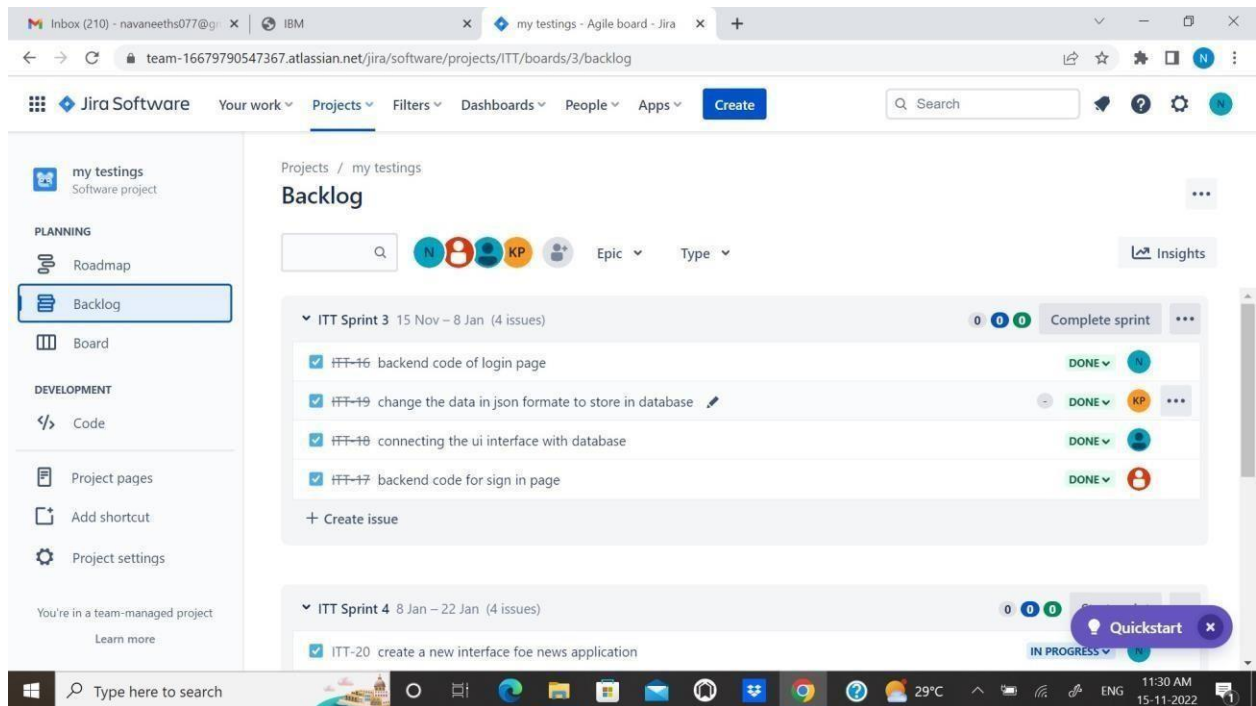
Sprint-1



Sprint-2

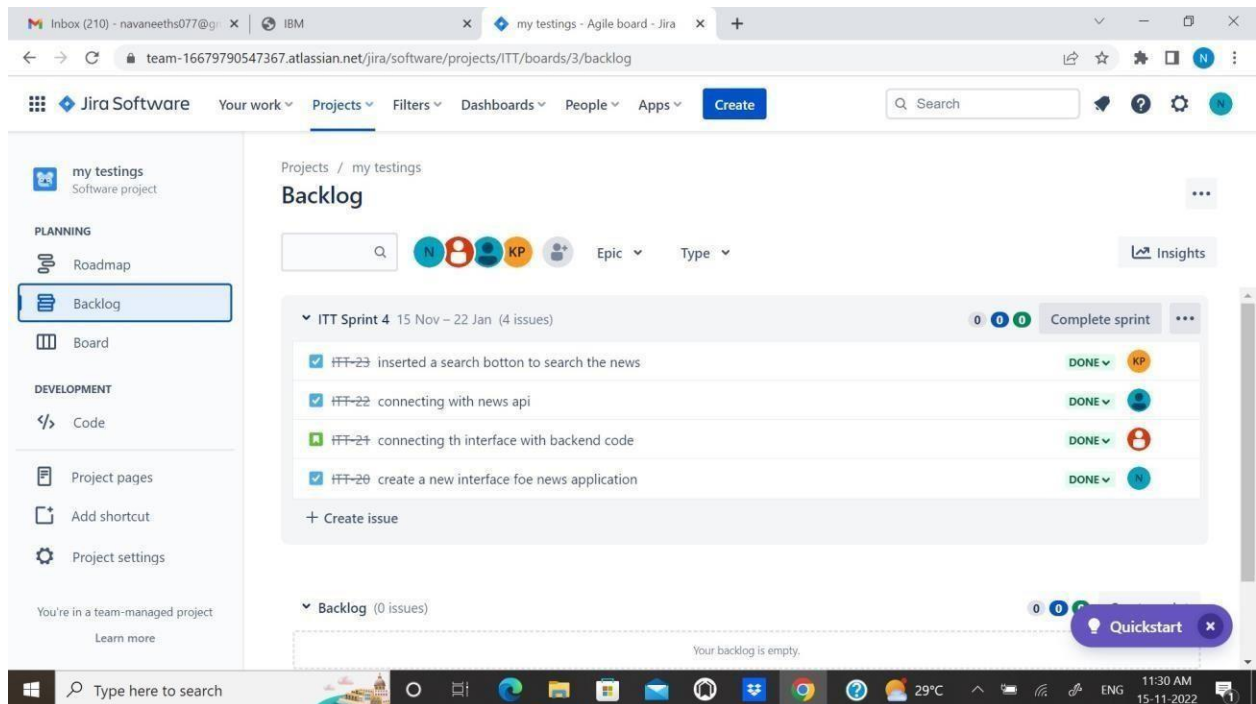


Sprint-3



The screenshot shows the Jira Software interface for the 'my testings' project. The left sidebar contains navigation options: my testings (Software project), PLANNING (Roadmap, Backlog, Board), DEVELOPMENT (Code), Project pages, Add shortcut, and Project settings. The main area displays the 'Backlog' for 'my testings'. It shows two sprints: 'ITT Sprint 3' (15 Nov – 8 Jan, 4 issues) and 'ITT Sprint 4' (8 Jan – 22 Jan, 4 issues). The 'ITT Sprint 3' backlog contains four issues, all marked as 'DONE' with green checkmarks and assigned to team members (N, B, KP, and another N). The issues are: #TT-46 (backend code of login page), #TT-19 (change the data in json formate to store in database), #TT-18 (connecting the ui interface with database), and #TT-17 (backend code for sign in page). The 'ITT Sprint 4' backlog contains one issue, 'ITT-20 create a new interface foe news application', which is marked as 'IN PROGRESS' and assigned to team member N. A 'Quickstart' button is visible in the bottom right corner of the sprint view.

Sprint-4



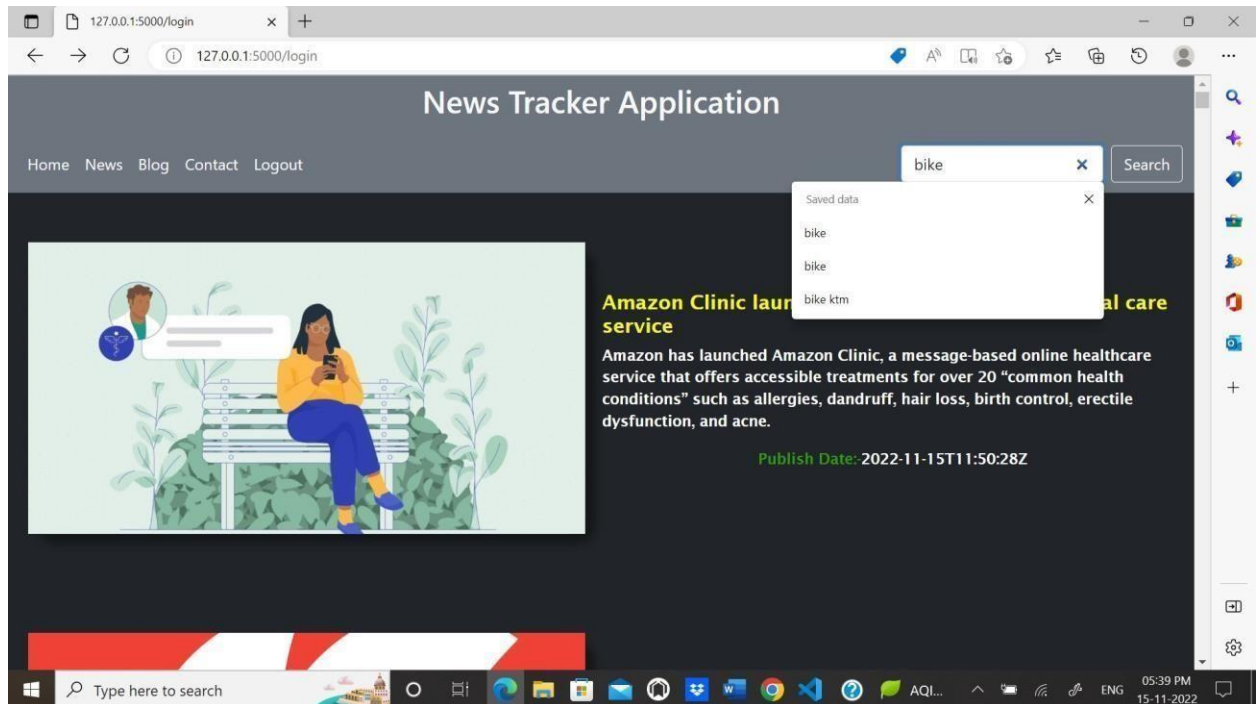
The screenshot shows the Jira Software interface for the 'my testings' project, displaying the 'Backlog' for 'ITT Sprint 4' (15 Nov – 22 Jan, 4 issues). The left sidebar is the same as in the previous screenshot. The main area shows the 'ITT Sprint 4' backlog with four issues, all marked as 'DONE' with green checkmarks and assigned to team members (N, B, KP, and another N). The issues are: #TT-23 (inserted a search botton to search the news), #TT-22 (connecting with news api), #TT-21 (connecting th interface with backend code), and #TT-20 (create a new interface foe news application). Below the sprint view, there is a section for the 'Backlog' (0 issues) which is empty, with the text 'Your backlog is empty.' displayed. A 'Quickstart' button is visible in the bottom right corner of the sprint view.

7.CODING & SOLUTIONING

7.1Feature 1: Verification email Sender:

```
1 def emailSender(email, token):
2     configuration = sib_api_v3_sdk.Configuration()
3     configuration.api_key['api-key'] = app.data['mail_api_key']
4     api_instance = sib_api_v3_sdk.TransactionalEmailsApi(
5         sib_api_v3_sdk.ApiClient(configuration))
6     now = datetime.now()
7     dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
8     msg = {}
9     msg['Subject'] = "Verfiy your NewsTracker Account"
10    msg['From'] = {"name": "News Tracker Dev Team",
11                  "email": "verify@news
12 tracker.com"}
13    msg['To'] = [{"email": email}]
14    msg['Text'] = f'Please click this <a href="http://127.0.0.1:5500/frontend/pages/verify.html?token={token}">link</a> to
15 verify your account'
16    html = f"""
17 <html>
18 <head></head>
19 <body>
20 <p>👋, for joining NewsTracker
21 r 🎉</p>
22 <br>
23 <p>Hurray🎉, you just registered
24 at NewsTracker<br><br>
25 Please click the following link
26 to verify your account:<br>
27 <a href="http://127.0.0.1:5500/frontend/pages/verify.html?token={token}">Click Here to Verify 🎉</a>
28 </p>
29 <br>
30 <p>⚠️Note: This link expires within one hour from the time sent</p>
31 <br><br>
32 <p>Regrads,<br></p>
33 <p><a href="https://localhost:5000">NewsTracker Dev Team</a></p>
34 <br><br>
35 <p>Email sent at {dt_string}</p>
36 </body>
37 </html>
38 """
39    send_smtp_email = sib_api_v3_sdk.SendSmtpEmail(
40        to=msg['To'], html_content=html,
41        sender=msg['From'], subject=msg['Subject'], text_content=msg['Text'])
42    try:
43        api_response = api_instance.send_transactional_email(send_smtp_email)
44        print(api_response)
45    except ApiException as e:
46        print("Exception when calling SMTPApi->send_transactional_email: %s\n" % e)
```

The above function is used to send the verification code to the desired email



7.2 Feature 2:

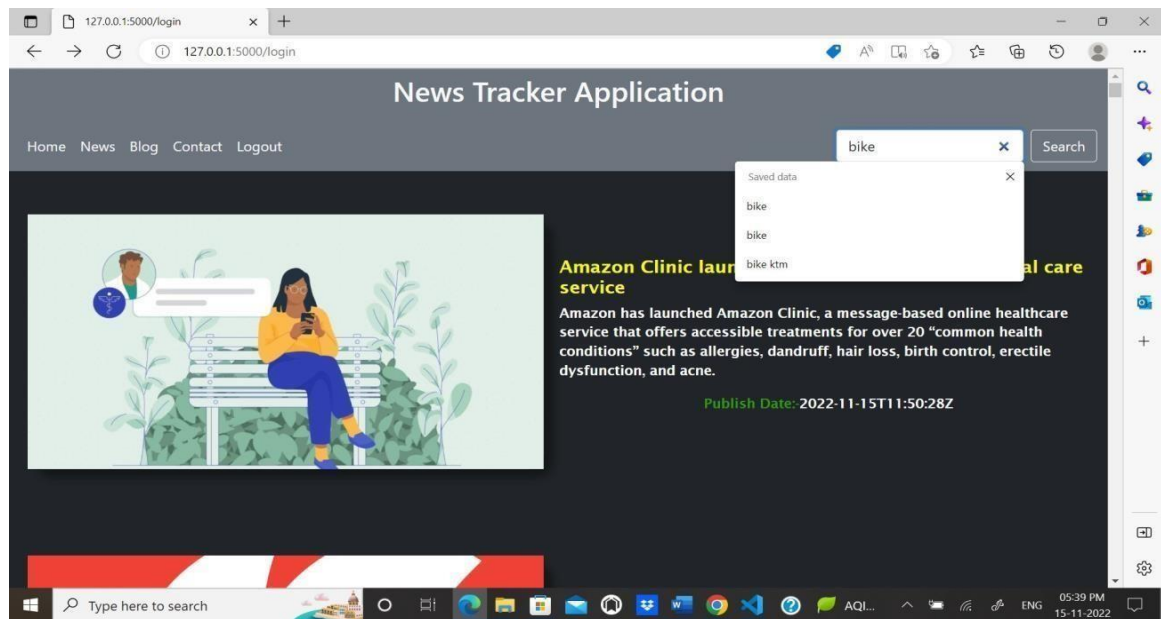
Cookie Checker:

```

1 def token_required(f):
2     @wraps(f)
3     def decorated(*args, **kwargs):
4         token = request.cookies.get("access_token")
5         try:
6             data = jwt.decode(token, app.app.config['SECRET_KEY'], algorithms=['HS256'])
7             ip=request.headers.get("ip")
8             cookieIp=data['ip']
9             if(ip!=cookieIp):
10                 resp={"status":"not logged in"}
11                 @after_this_request
12                 def deleter(response):
13                     response.delete_cookie("access_token",path="/")
14                     response.delete_cookie("email",path="/")
15                     return response
16                 return resp,401
17         except:
18             resp = {"status":"not logged in"}
19             @after_this_request
20             def deleter(response):
21                 response.delete_cookie("access_token",path="/")
22                 response.delete_cookie("email",path="/")
23                 return response
24             return resp, 401
25         return f(data['email'],*args, **kwargs)
26     return decorated

```

This code is used to check the cookie from the client side and checks whether the user is signed in or not.



7.4 Database Schema:

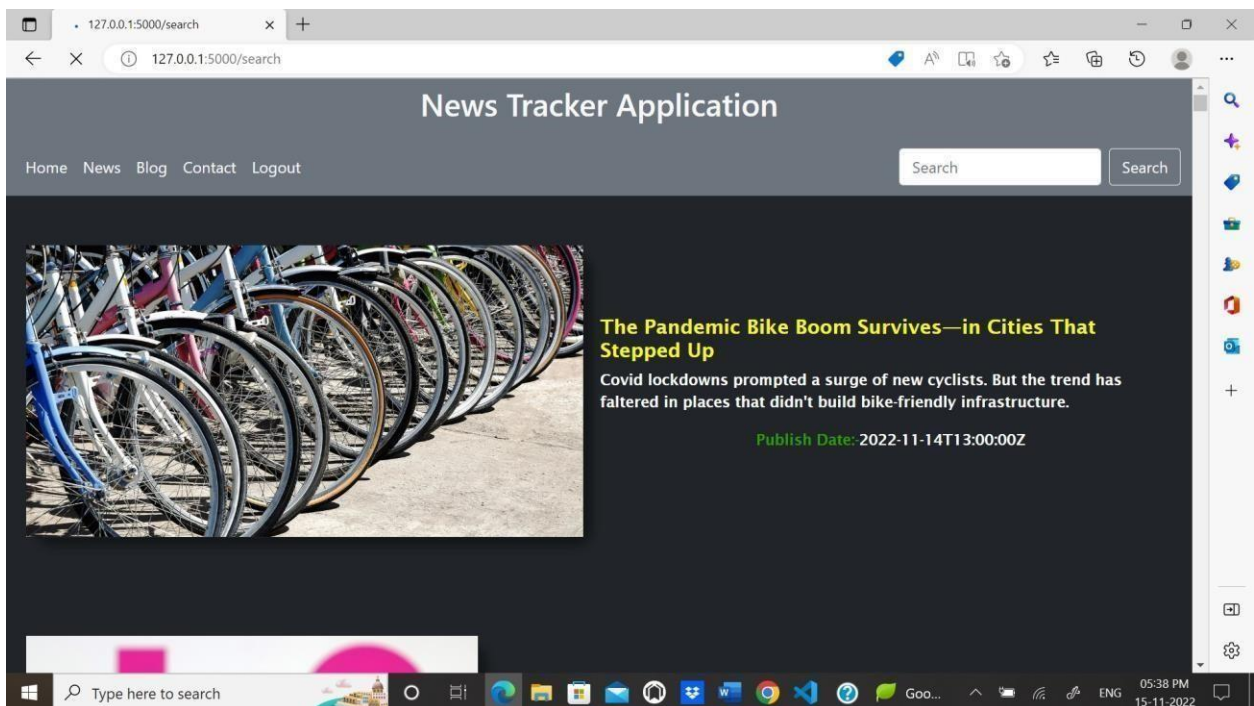
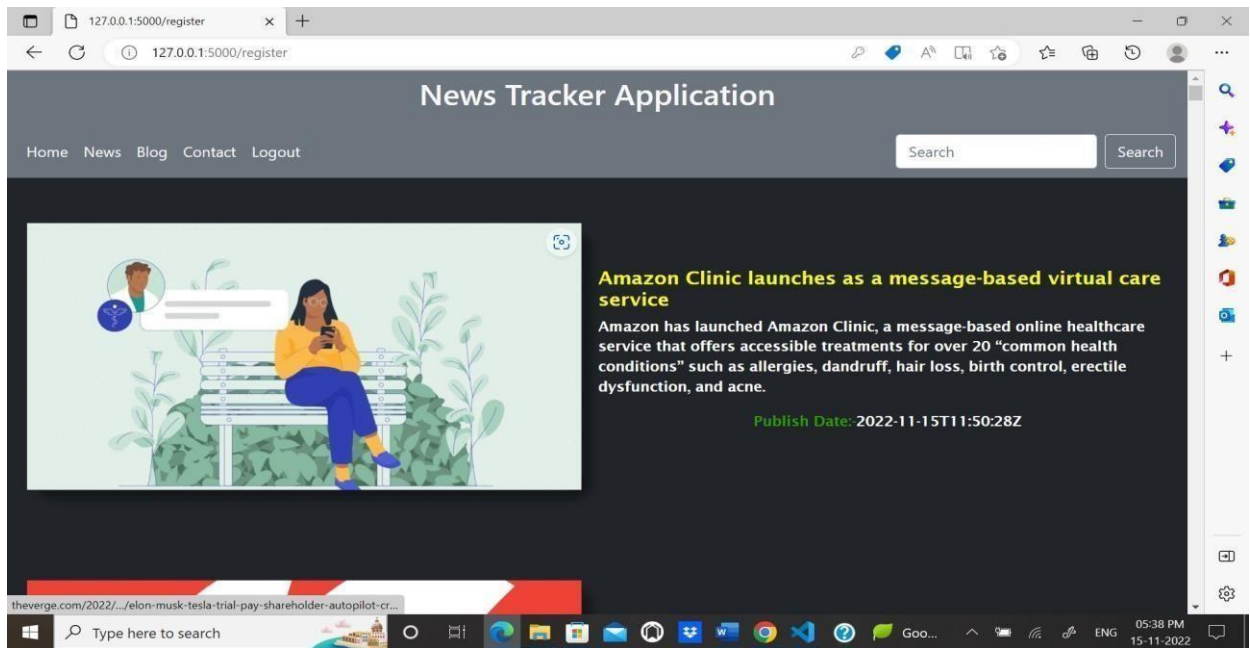
User Table

Table definition

USER

Approximate 2 rows (4.03 MB)
Updated on 2022-10-17 15:43:28

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	👁
NAME	VARCHAR	N	255	0	👁
EMAIL	VARCHAR	N	255	0	👁
PASSWORD	VARCHAR	N	255	0	👁
FAVOURITES	CLOB	Y	1048576	0	👁
BOOKMARKS	CLOB	Y	1048576	0	👁
VERIFIED	BOOLEAN	Y	1	0	👁
RESEND_TIME	VARCHAR	N	255	0	👁



8.TESTING

8.1. User Test case:

/

Test case	Feature	Components	Test scenario	Expected result	Actual result
Sign in	Functional	Login page	Verify user can see sign in option	Can visible	Yes visible
Sign up	Functional	Login page	Verify user can sign up	Can visible	Yes visible
Forgot news	Functional	Login page	Verify user can see forgot password	Option is available	Option is available
Fetch news	Functional	Home page	Verify user can see the news	News will be feed to the app	404 error
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.

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues [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 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

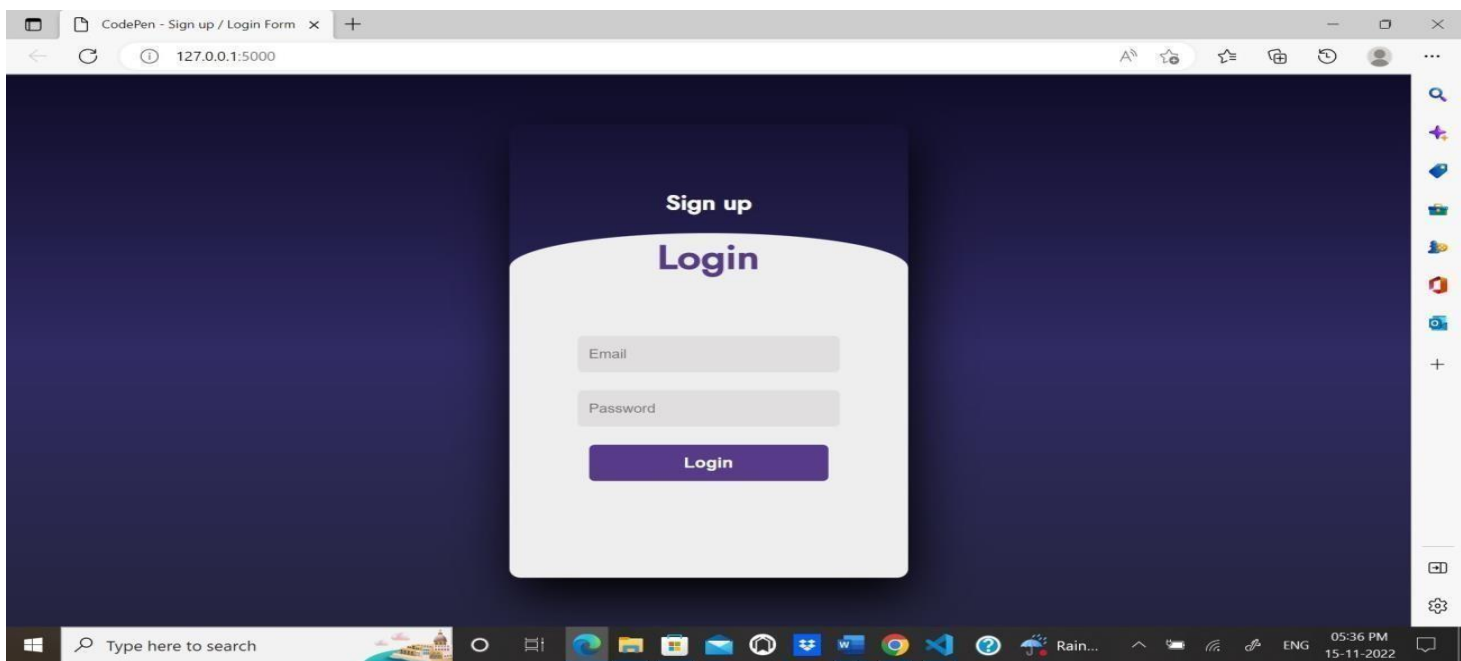
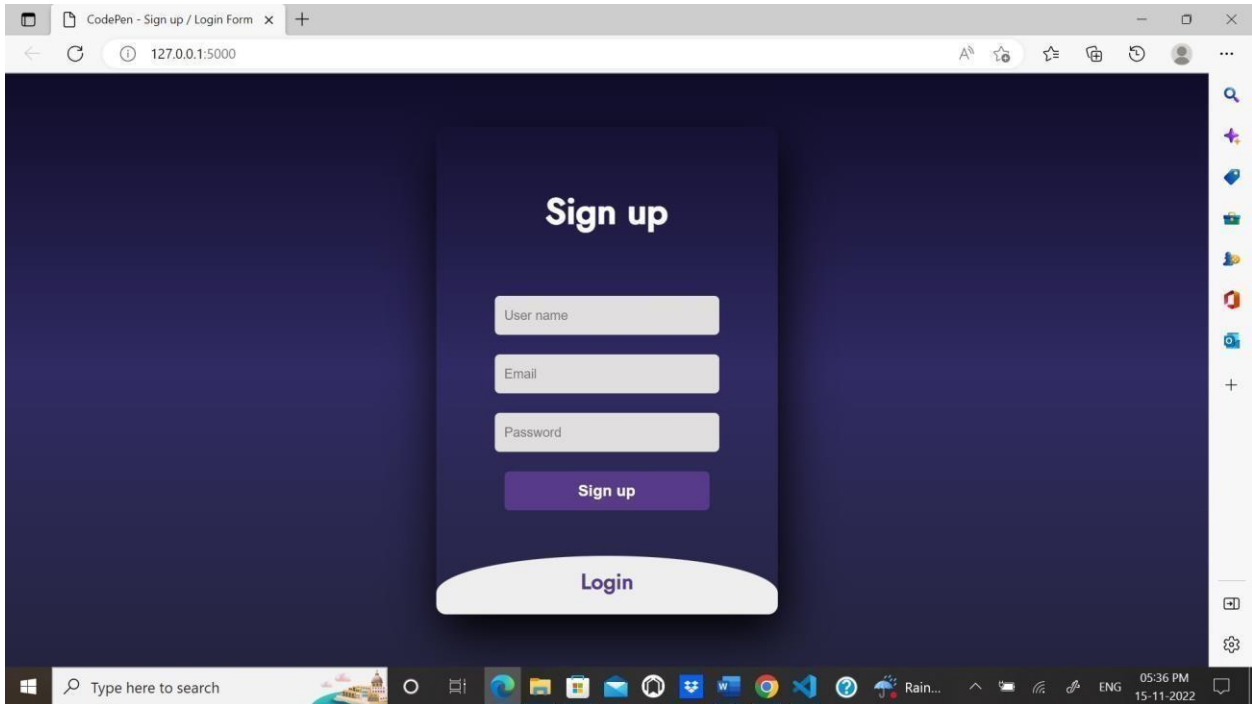
3. Test Case Analysis

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 Result :

9.1 Performance 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.
4. With News tracking , **you get insights into the performance of your business**. News or media coverage, whether online or traditional, gets your message to a wider range of audiences and promotes brand awareness of your business. Being featured at an independent media like News gives credibility to the information.
- 5.And by tracking what kind of credibility you have formed, you can work to bring the credibility you wish to build. Maintaining positive news increases positive credibility. This app can be accessed anywhere and anytime, So that the user can view the news Its ad free . The news is only based on the API

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.
 - 3.It CAN BE limited by time.
 - 4.It may rely too heavily on personalities, emotions, opinions... not facts.
 - 5.It can shortchange complex stories or avoid them altogether.
- - 6. Difficult To Implement Effectively. Fast-tracking is one of the more difficult strategies to implement in the workplace. ...
 -
 - 7.Requires A High Level Of Project Understanding. ...
 - 8.High Risk Of Low-Quality Work. ...
 -
 - 9.Increased Stress. ...
 -
 - 10.Risk Of Unexpected Costs.

11.CONCLUSION

Thus, the projected using IBM CLOUD is tested, verified and executed successfully. Thus we have developed a full stack application based on the plans and within the given time. We have tested the application in both desktop and mobile and it worked well, Overall it was a great experience

12.Future scope

In future, we planned to implement this project in large scale which will be helpful and used by all the people. In future we may integrate our own news API instead of third party APIs and may develop a mobile native application so that it can be used in both android and ios.

13.APPENDIX

[Source Code \(Github\)](#)

[Live Page URL](#)

[Video Demo Link](#)

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">Home

<li class="nav-items">News

<li class="nav-items">Blogs

<li class="nav-items">Contact

 -->

</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-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"

ariaexpanded="false" arialabel="Toggle navigation">

</button> -->

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

Home

<li class="nav-item">

News

<li class="nav-item">

Blog

<li class="nav-item" role="contact">

Contact

<li class="nav-item" role="logout" >

Logout


```

        <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-

```



```
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>
```

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>
            <th scope="col">Contact (Click the Symbol) </th>
        </tr>
        <tr>

            <td>Naveen Raja N</td>
            <td>

                <a href="https://www.linkedin.com/in/naveen-raja-n-5154711b3"><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-
    1.3421.248-.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
    1.634v3.865h2.401V9.25c02.221.1843.252-2.764-3.252-1.274 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>Navaneeth S</td>

```

```

    <td>

```

```

    <a href="https://www.linkedin.com/in/navaneeth-s-706b40214"><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-
    1.3421.248-.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
    1.634v3.865h2.401V9.25c02.221.1843.252-2.764-3.252-1.274 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>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-
        1.3421.248-.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
        1.634v3.865h2.401V9.25c02.221.1843.252-2.764-3.252-1.274 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>Karuppiah AL </td>

```

```

    <td>

```

```

        <a href="https://www.linkedin.com/in/karuppiah-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-
        1.3421.248-.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
        1.634v3.865h2.401V9.25c02.221.1843.252-2.764-3.252-1.274 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+LWgxfKTRIcfu0JTxR+EQDz/bglIdoEyl4H0zUF0QKkbrJ0EcQF"
crossorigin="anonymous"></script>

```

```

</body>

```

</html>

Style.css body

```
{
  margin: 0px;
  padding: 0px; background-color:
  rgb(69, 54, 54);
}
* { box-sizing: borderbox;
} 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;
  justifycontent: space-around;
  alignitems: 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; textdecoration: none
!important; letterspacing: 1px;
}

nav .menu { display:
flex; flexflow: row
wrap; justifycontent:
center; alignitems:
center;
} nav .menu li a { height: 35px;
line-height: 30px; display:flex ;
texttransform: uppercase;
align-items: center;
padding: 0px 22px; fontsize:
```

```

15px; font-weight: bold;
    color: white; fontfamily:
    'Poppins', sans-serif; letterspacing:
    1px;
}
nav .menu li a:hover { color:
black; backgroundcolor:white;
border-radius:
20px;
}
#main {
    /* padding-top: 98px; */
}
.banner { height: 620px;
width: 100%; display: flex;
flex-wrap: column;
justifycontent: center;
alignitems: center;
backgroundcolor: black; border:
1px solid rgb(98, 4, 4); }
.banner img { height: 100%;
width: 100%; padding: 2px; }
.headgit{ height: 40px; width:
100%; justifycontent: 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 { textdecoration:
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;
justifycontent: center;
color: white; font-size:
30px; fontweight: bold;
fontfamily:'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: spacebetween;
/* background-color: rgb(68, 47, 47); */
}
.wrapper .item { height: 300px; min-width: 30%;
border-radius: 20px; margin: 10px; boxshadow:
10px 10px 10px rgba(0, 0, 0, 0.4); padding: 0.5px;
flex-flow: column wrap; borderbottom: 1px white;
} .wrapper .item img
{ height: 100%; width: 100%; }
.c-box { height: content;
max-width: 50%; display:
flex; align-items: center;
padding: 10px; flexflow:
column; padding: 1px;
}
.c-box .content { height:
content; minwidth: 30%;
display: flex; align-items:
center; flex-flow: row wrap;
}
.c-box .content p {
color: #fff; fontsize:
15px; fontweight: 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%; justifycontent:
center; alignitems: 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;
justifycontent: center; alignitems:
center; min-height: 100vh; fontfamily:
'Jost', sans-serif; background:
lineargradient(to bottom, #0f0c29,
#302b63, #24243e);
}
.main{ width: 350px; height:
500px; background: red;
overflow:
hidden;
background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dmis5/fo0picsp1nhiucmc
0l2
5s29respgpr4j/1631524275000/03522360960922298374/03522360960922298374/1Sx0jhdpEpnNIydS4 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;
fontweight: bold; cursor: pointer;
transition: .5s ease-in-out; }
input{ width: 60%; height: 20px;
background: #e0dede;

```



```

justifycontent: center; display:
flex; margin: 20px auto; padding:
10px; border: none; outline: none;
borderradius: 5px; } button{
width:
60%; height: 40px; margin:
    10px auto;
    justifycontent:
    center; display: block; color:
    #fff; background:
    #573b8a; font-size:
    1em; font-weight: bold;
margin: 20px 0 20px 0; border:
none; borderradius: 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-
abf121c6f737b08c-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-51220-1660976139>

Project Demo Link: https://drive.google.com/file/d/1n0rKi_zmliEqf2i7FbzWHt8fNHRXSyJ/view?usp=drivesdk

REFERENCE VIDEO : <https://vimeo.com/742609168/1824d26a5b>