

IBM-Project-7918-1658902616

NEWS TRACKER APPLICATION

Project Domain : Cloud Application Development

Team ID : PNT2022TMID37337

TEAM MEMBERS

1.Vijai

2.Gokul

3.Devanand

4.Gokul Raj

1. INTRODUCTION

Project Overview

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 is not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you are 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.

1.2 Purpose

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every minute and distributed via television, radio, and the Internet, so the news updated the next morning is already outdated.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.

References

1. <https://flask.palletsprojects.com/en/2.1.x/tutorial/>
2. <https://www.ibm.com/docs/en/db2-for-zos/11?topic=zos-db2-basics-tutorial-working-db2-interactively>
3. <https://developer.ibm.com/components/cloud-foundry/tutorials/>
4. <https://www.ibm.com/in-en/products/watson-assistant/docs-resources>
5. <https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-getting-started-cloud-object-storage>

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. Creating a user-friendly news app that would not just have good number of features, but will also be able to get accessed by different types of users keeping the experience delightful.

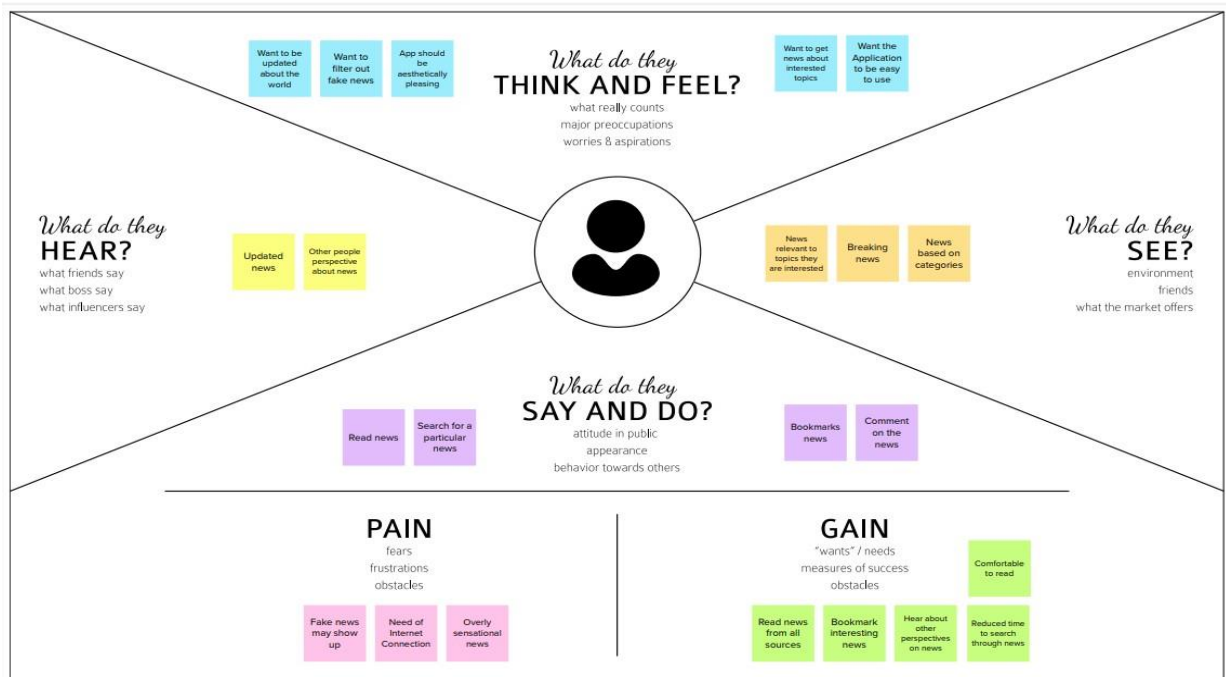
S.No	Topic	Author	Abstract	Year	Accuracy
1.	Opinion Extraction, Summarization and tracking in News	Lun Wei Ku, Yu Ting Liang, Hsin-hsi Chen	TREC, NTCIR and articles collected from web blogs and news serve as the info sources of opinion extraction. Algorithms for opinion extraction at word, sentence and document level are proposed. The issue of relevant sentence selection is discussed, and then topical and opinionated information are summarized and visualized by representative sentences.	2003	85%
2.	News Classification	Mazhar Iqbal Rana, Sherzad Khalid, Muhammad Usman Akbar	The proposed system focuses on full news, huge documents, long length texts etc. It is more prominent as compared to the short length text. Text classification process, classifiers, and numerous feature extraction methodologies but all in context of short texts i.e. news classification based on their headlines. Existing classifiers and their working methodologies are being compared and results are presented effectively.	2014	76%

3.	Breaking News Detection and Tracking in Twitter	Swit Phuvioadawat, Tsuyoshi Murata	<p>Mainly Twitter has been used as one of the communication channels for spreading breaking news.</p> <p>We propose a method to collect, group, rank and track breaking news in Twitter. Since short length messages make similarity comparison difficult, we boost scores on proper nouns to improve the grouping results based on popularity and reliability factors.</p>	2010	78%
4.	Tracking News stories across different sources	Yun Zhai, Mubarak Shah	<p>Tracking method in which links news stories on the same topic across multiple sources. The semantic linkage between the news stories is reflected in combination of both their visual content and their spoken language content.</p> <p>Visually, each news story is represented by a set of key-frames with or without detected faces. The facial key-frames are linked based on the analysis of the extended facial regions, and the non-facial key-frames are correlated using the global Affine matching.</p>	2005	80%

5.	Learning approaches for detecting and tracking news events	Y. Yang, J.G. Carbonell, R.D Brown, T. Peirce, B.T. Archibald	<p>The article discusses our research using manually segmented documents. It is used supervised learning and unsupervised clustering algorithms to allow document classification based on the information content and temporal aspects of news events. They've adapted several IR and machine learning techniques for effective event detection and tracking.</p>	1999	60%
----	--	---	---	------	-----

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



Ideation & Brainstorming



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 🕒 10 minutes to prepare
- 🕒 1 hour to collaborate
- 👥 2-6 people recommended



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

- A** Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
- B** Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
- C** Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →



Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

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. 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, for filters like Breaking News, Most Popular, and more. Creating a user-friendly news app that would not just have good number of features, but will also be able to get accessed by different types of users keeping the experience delightful.



Key rules of brainstorming

To run a smooth and productive session

- 😊 Stay in topic.
- 💡 Encourage wild ideas.
- 🙊 Defer judgment.
- 👂 Listen to others.
- 🗣️ Go for volume.
- 👁️ If possible, be visual.



Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Kessav

Easy to maintain	Interactive UI	Maintain Database
Small app size	Modify News	Personal preference
IBM Cloud	News Sharing	Write maintainable code

Madhumitha

Hash user credentials	Add to favourites	Signup
Speed reading option	Different language options	Automation of App
Better Security & Backup	IBM DB	Faster Data Fetching

Mugunthan

Responsive UI	Reliable news	Finding genuine news
Neglect irrelevant news	Security	No Ads
User data Privacy	Regular notifications	Regular Updates

Naren

Less distracting	24/7 Availability	User can add their own news
Lightweight	Overcomes copyright issues	Accurate Results
Reduced spam notifications	User friendly	Quick authentication

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

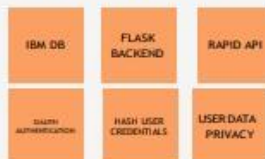
UI / UX



APP FEATURES



BACK END



MAINTAINABILITY

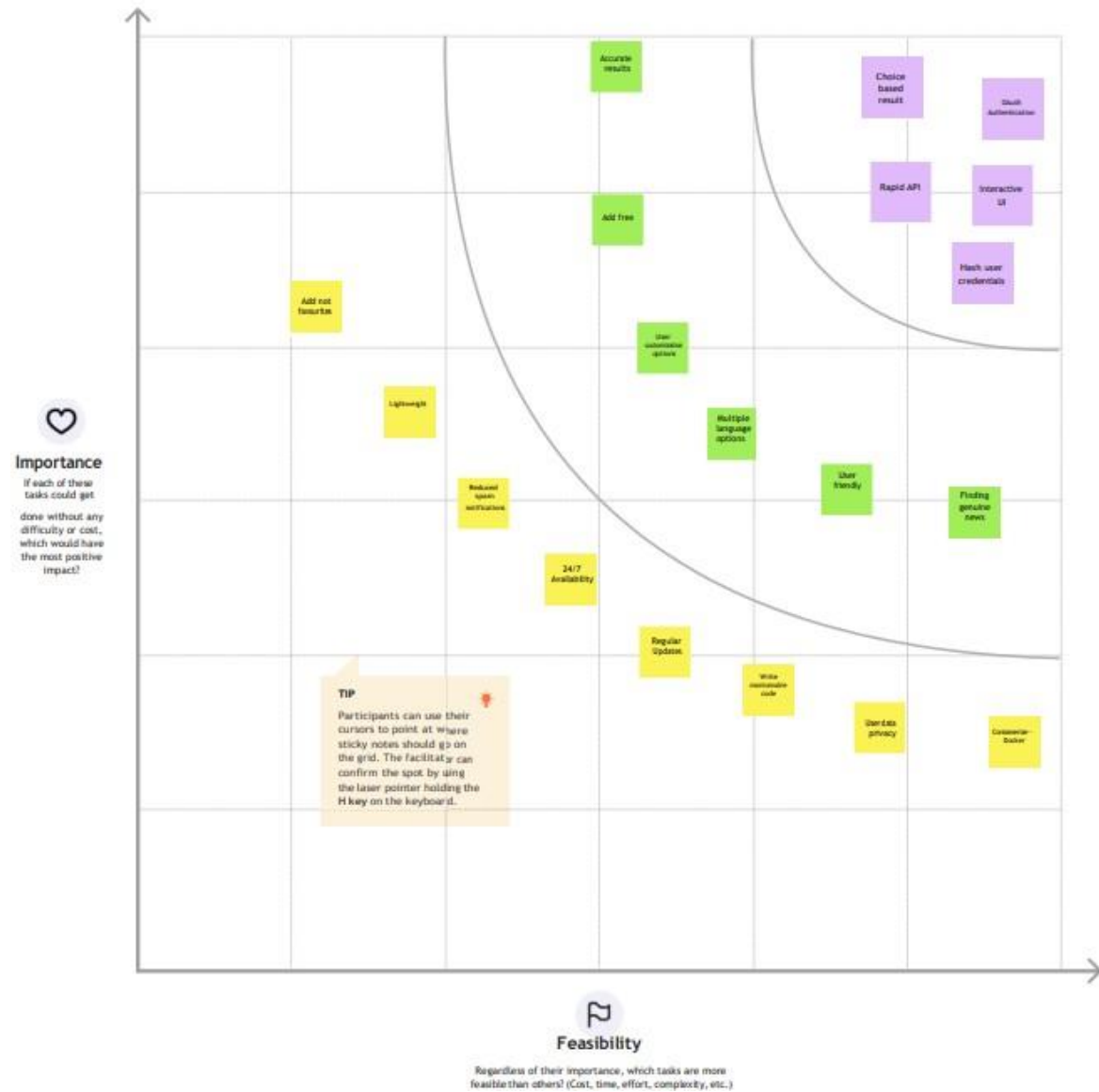


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes



Proposed Solution

S.NO.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Statement: Everyday, a lot of events happen world-wide and we rely on newspapers, television and news articles to get the reliable and trust-worthy information about these events.</p> <p>Description: As a result, we created a platform that offers such news from reliable sources worldwide, in an organized and efficient manner.</p>
2.	Idea / Solution description	<p>One platform for all local and worldwide news. Trustworthy and Reliable News.</p> <p>Fast and efficient system.</p> <p>Preventing spread of False information.</p> <p>Data Storage and Backup.</p> <p>Communication.</p>
3.	Novelty / Uniqueness	<p>A cloud computing-based news application that generates news and reports about the happenings around the world using computers and network (Internet).</p> <p>News based on most reliable and trustworthy resources around the world.</p> <p>Developing the Eco-Friendly & sustainability based centre.</p>

4.	Social Impact / Customer Satisfaction	<p>Cloud computing offers a way to create, coordinate, and share information across the globe.</p> <p>The adoption of cloud-based services gives access to a wider range of data and sharing the important information in an efficient way.</p> <p>Our platform eliminates the spread of false news and exposes the injustice and wrongdoings done by false groups.</p> <p>Eliminating the fake news provides better understanding of the real-events happening in the world and the spread of knowledge.</p>
----	---------------------------------------	---

5.	Business Model (Revenue Model)	<p>Our application covers a range of topics including politics, business, criminal justice, environment, technology etc.</p> <p>Our business model will be monetized and generate income by showing advertisements and Operating on monthly and yearly subscription model.</p>
6.	Scalability of the Solution	<p>Scalability is one of the benchmarks of the cloud services and its adoption with businesses.</p> <p>Cloud scalability will help to increase the user-base by increasing the resource allocation and meeting the changing demands without sacrificing the efficiency or quality of our customer service and internal operations.</p> <p>Providing fast and reliable news while maintaining positive relationships with your customers.</p>

Problem Solution fit

Project Title: News Tracker Application

Project Design Phase-I - Problem Solution Fit

Team ID: PNT2022TMID20111

DefineCS, hinttoCC	1.CUSTOMER SEGMENT(S) <p>Hackers,CCTNS(crime and criminal tracking network and system),commercial and scientific purposes,media monitoring.</p>	6.CUSTOMER CONSTRAINTS <p>As much as service providers need to cater to the needs of their customers, it is just as important for them to satisfy their customers.</p>	5.AVAILABLE SOLUTIONS <p>The internet and the intermingling of social media with important worldwide events has made it almost impossible to live under a rock. But finding a reliable one-stop shop to engage with your news can be somewhat of a challenge.</p>	ExploreAS differentiate
	2.JOBS-TO-BE-DONE / PROBLEMS <p>We've written a whole white paper on the topic ,and examples of how our stories have produced such change-from the registration. It corrupt officials to the passage of news laws- are compiled in our annual reports, on this page you'll find our reporting on the impact of our work.</p>	9.PROBLEM ROOT CAUSE <p>User doesn't want to waste time figuring out the the relevance of the news young people don't read news from apps, usually depend on social media to get updated Users sometimes get overwhelmed by too many categories.</p>	7.BEHAVIOUR <p>User experience,content performance popularity sharing the news, conversion rate optimization ,checkout process website accessebility,livechat and first and foremost,user data is super important.</p>	
IdentifystrongTR&EM	3.TRIGGERS <p>This news app is a big interactive database that think of it like would any other piece of journalism .It just uses software instead of words and pictures.</p>	10. YOUR SOLUTION <p>Realtime monitoring ,working condition is ensured and restricted permission from entering highly secured areas.</p>	8.CHANNELS of BEHAVIOURS 8.1 ONLINE <p>A few of these local stories were the result of formal partnerships, but the majority were done quite independently – in some cases, we didn't have much if any knowledge that the story .</p> 8.2 OFFLINE <p>Although marketing has gone largely digital, the offline advertising world is still a significant one, means a lot of money gets spent on display ads.</p>	Extractonline&offlineCh&BE
	4.EMOTIONS: BEFORE /AFTER <p>Fearness,statsatisfaction,anger and two sentiments(positive and negative)using to extraction.</p>			

4.REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution

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

Non-functional Requirements:

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

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

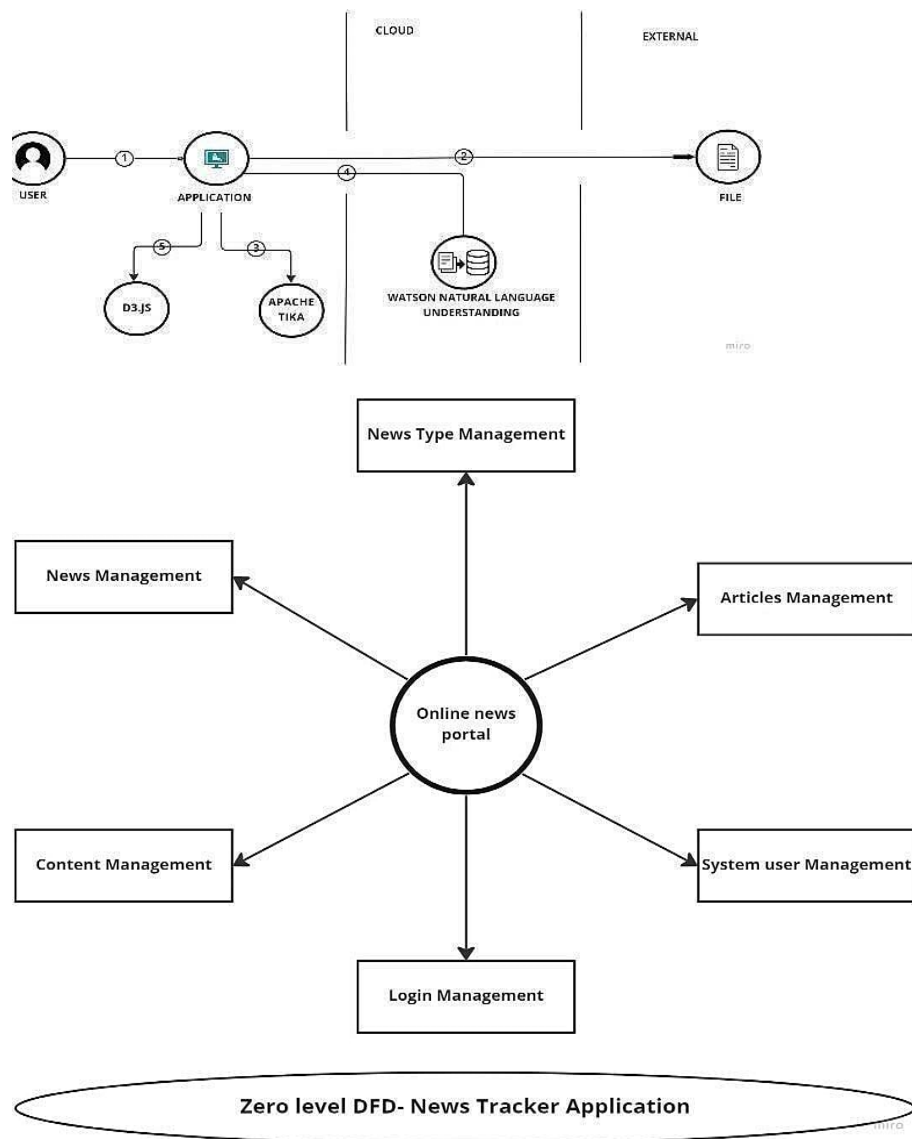
5.

PROJECT DESIGN

Data Flow Diagrams

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

Example: (Simplified) Flow

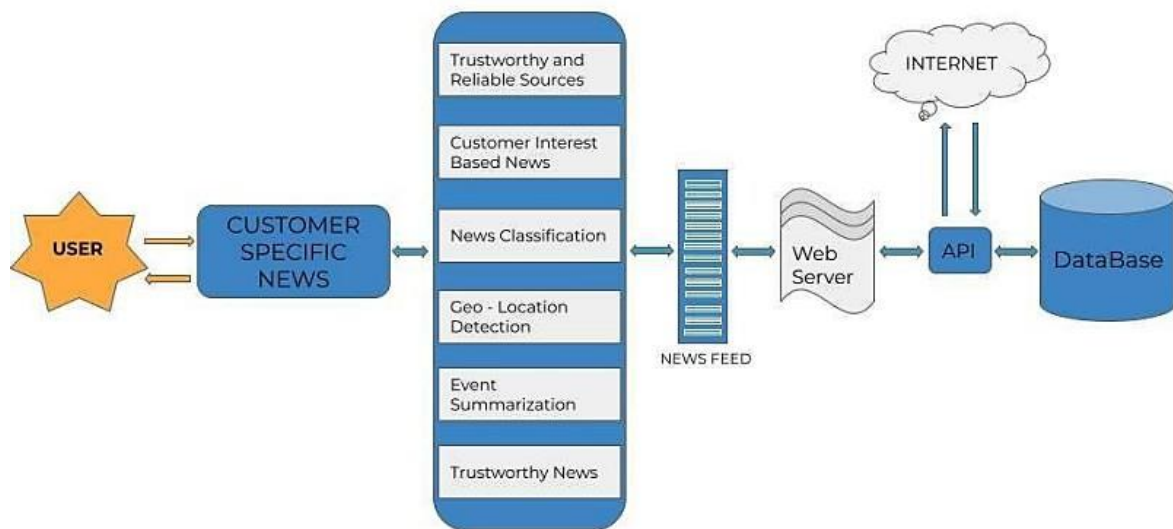


Solution & Technical Architecture

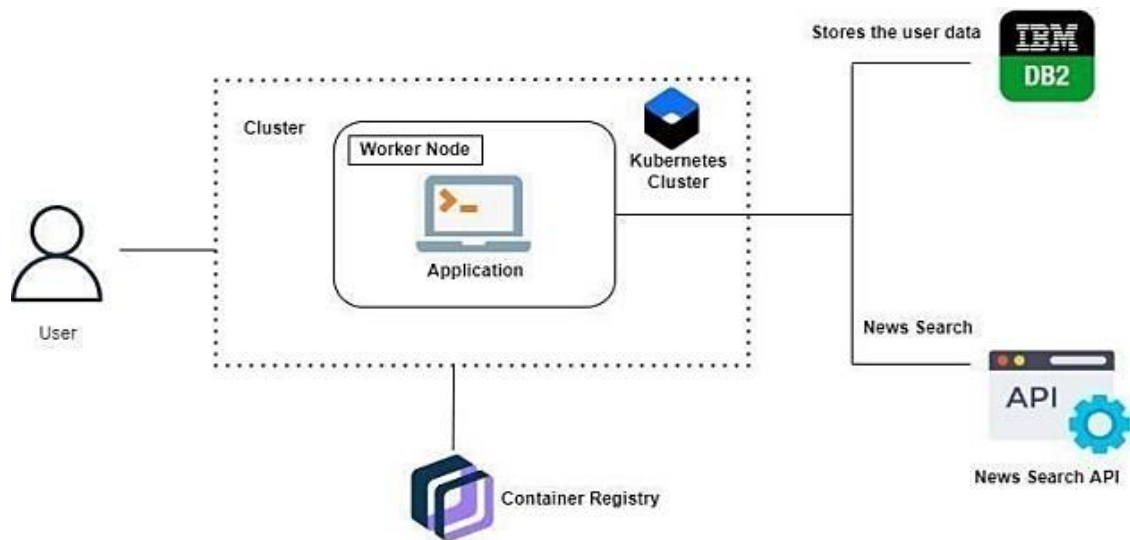
Solution architecture is a complex process, with many sub-processes that the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems. Describe the structure, characteristics, behavior, 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.

Solution Architecture diagram



Technical Architecture



User Stories

Phases	Purchasing a device				Downloading the application			Create an account		Choosing the necessary genre		
Steps	Make sure we have the required money	Android or iOS	Payment mode		Good internet connectivity	Gmail account or Apple id	Enough storage in the device	Login credentials	Mail ID	Different sign-in modes	Preferred Language	Area of interest
Feelings	Excited to have new device				Eager how well this works			Successfully created			Reading based on the interest	
	Is the device affordable?				What if the space and data is insufficient			Will my information be secured?			Unwanted notifications	
Pain points	Is the device affordable?				What if the data and space is efficient			Will my information be secured?			Unwanted notifications	
Opportunities	Regularly checking on newly launched devices				Delete unwanted files and proper wifi			Double check privacy and policy			Follow only the required genre	

6.PROJECT PLANNING & SCHEDULING

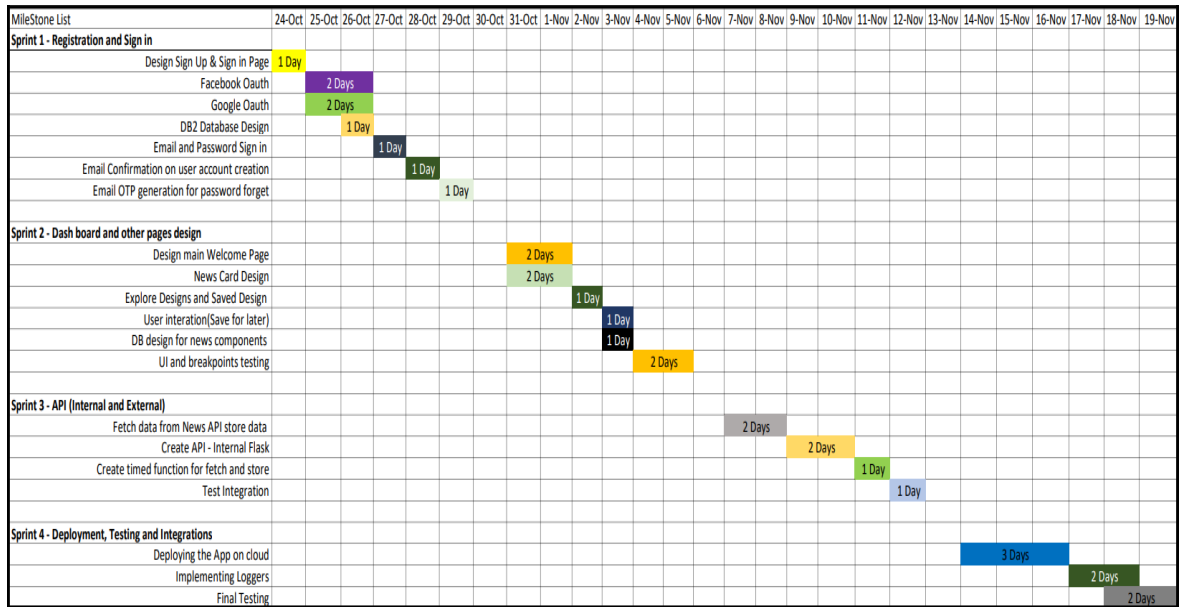
Sprint Planning & Estimation

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	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Mugunthan Kessav Madhumitha Naren shanmugam

Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-3	Dashboard	USN-6	As a user, I will able to see the login and profile details	2	Medium	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-4	Chat box	USN-8	As a customer care executive, I can ask about the inconvenience and the usability about the app	2	Medium	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-1	IBM cloud	USN-10	As a tracker, I collect all the NEWS from the cloud	1	Medium	Mugunthan Kessav Madhumitha Naren shanmugam
Sprint-3	IBM DB2	USN-11	As a database manager, I can collect all the user details and NEWS application details	2	Medium	Mugunthan Kessav Madhumitha Naren shanmugam

Sprint Delivery Schedule

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



Reports from JIRA

← → ↻ 19tuec108.atlassian.net/jira/software/projects/NTA/boards/1/backlog

Jira Software Your work Projects Filters Dashboards People Apps Create Search

NEWS TRACKER APPLI... Software project

PLANNING Roadmap

Backlog

Board

DEVELOPMENT Code

Project pages

Add shortcut

Project settings

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / NEWS TRACKER APPLICATION

Backlog

KB MR MR MR Epic

NTA Sprint 1 24 Oct – 29 Nov (5 issues) 0 0 0 Complete sprint

- NTA-1 As a user, I can register for the application by entering my email, password, and confirming my password. DONE KB
- NTA-3 As a user, I can register for the application through Gmail. DONE MR
- NTA-4 As a user, I can log into the application by entering email & password. DONE MR
- NTA-5 As a tracker, I collect all the NEWS from the cloud. DONE MR
- NTA-2 As a user, I will receive confirmation email once I have registered for the application. DONE MR

+ Create issue

Jira Software Your work Projects Filters Dashboards People Apps Create Search

NEWS TRACKER APPLI... Software project

PLANNING Roadmap

Backlog

Board

DEVELOPMENT Code

Project pages

Add shortcut

Project settings

Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / NEWS TRACKER APPLICATION

Backlog

KB MR MR MR Epic

NTA Sprint 2 31 Oct – 5 Nov (1 issue) 0 0 0 Complete sprint

- NTA-6 As a user, I can register for the application through Facebook. DONE MR

+ Create issue

NTA Sprint 3 7 Nov – 12 Nov (1 issue) 0 0 0 Complete sprint

- NTA-7 As a user, I will be able to see the login and profile details. DONE MR

+ Create issue

19tuec108.atlassian.net/jira/software/projects/NTA/boards/1/backlog

Jira Software

Your work

Projects

Filters





Dashboards

People

Apps


Create


Q Search




NEWS TRACKER APPLI...
Software project


PLANNING


 Roadmap


 Backlog


 Board

DEVELOPMENT

 Code

 Project pages

 Add shortcut

 Project settings


Does your team need more from Jira? Get a free trial of our Standard plan.


Projects / NEWS TRACKER APPLICATION


Backlog

KB

MR









Epic

Q


Insights


+ Create issue

NTA Sprint 4 14 Nov – 19 Nov (1 Issue)

0   Complete sprint

...

 NTA-B As a customer care executive, I can ask about the inconvenience and the usability about the app

DONE 

+ Create issue

Features

```

from flask import Flask, render_template, session, request, redirect, make_response
import ibm_db
from ibm_db2_connect import db2
import news
import json
app = Flask(__name__)
app.secret_key = "secret_key"
@app.route('/', methods=['GET', 'POST'])
def index():
    if session.get('logged_in'):
        return render_template('home.html')
        return render_template('index.html')
    @app.route('/login', methods=['POST'])
    def login():
        form_user = request.form['username']
        session['logged_in']=form_user
        return redirect('/')
    @app.route('/loginvalidate', methods=['POST', 'GET'])
    def loginvalidate():
        conn = db2.get_conn()
        data=request.get_json();
        form_user = data['email']
        form_password = data['password']
        sql = 'SELECT username from user WHERE mail=? and password=?'
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, form_user)
        ibm_db.bind_param(stmt, 2, form_password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            response = make_response('success',200)
        else:
            response = make_response('failure',200)
        response.mimetype = "text/plain"
        return response

```

```

@app.route('/register',methods=['POST'])
def register():
    try:
        conn = db2.get_conn()
        data=request.get_json();
        form_user = data['username']
        form_password = data['password']
        form_mail = data['mail']
        form_phone = data['phone']
        sql = 'insert into user values(?,?,?,?)'
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, form_user)
        ibm_db.bind_param(stmt, 2, form_password)
        ibm_db.bind_param(stmt, 3, form_mail)
        ibm_db.bind_param(stmt, 4, form_phone)
        ibm_db.execute(stmt)
        response = make_response('success',200)
    except Exception as e:
        response = make_response('failure',200)
    print(e)
    return response

@app.route('/checkForExistingUser',methods=['POST'])
def checkForExistingUser():
    conn = db2.get_conn()
    data=request.get_json();
    email=data['email'];
    print(email)
    sql = 'SELECT mail from user WHERE mail=?'
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        response = make_response('true',200)
    else:
        response = make_response('false',200)
    response.mimetype = "text/plain"
    return response

```

```
@app.route('/logout')
def logout():
    session.clear()
    return render_template('index.html')
if __name__ == '__main__':
    app.run(debug=True, port=5002)
```

```
import ibm_db
import config as c
class db2:
    connectionString="DATABASE={0};HOSTNAME={1};PORT={2};SECURITY={3};SSLServerCertificate={4};U
ID={5};PWD={6}".format(c.DATABASE,c.HO
conn = ibm_db.connect(connectionString,"")
def get_conn():
    return db2.conn
#NEWSTRACKER APPLICATION_TEAM_IDPNT2022TMID20111
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8" />
<title>Login and Registration in HTML | CodingNepal</title>
<link rel="stylesheet" href="../static/index.style.css" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
<body>
<div class="wrapper">
<div class="title-text">
<div class="title login">Login</div>
<div class="title signup">Signup</div>
</div>
<div class="form-container">
<div class="slide-controls">
<input type="radio" name="slide" id="login" checked />
<input type="radio" name="slide" id="signup" />
<label for="login" class="slide login">Login</label>
<label for="signup" class="slide signup">Signup</label>
<div class="slider-tab"></div>
</div>
<div class="form-inner">
```

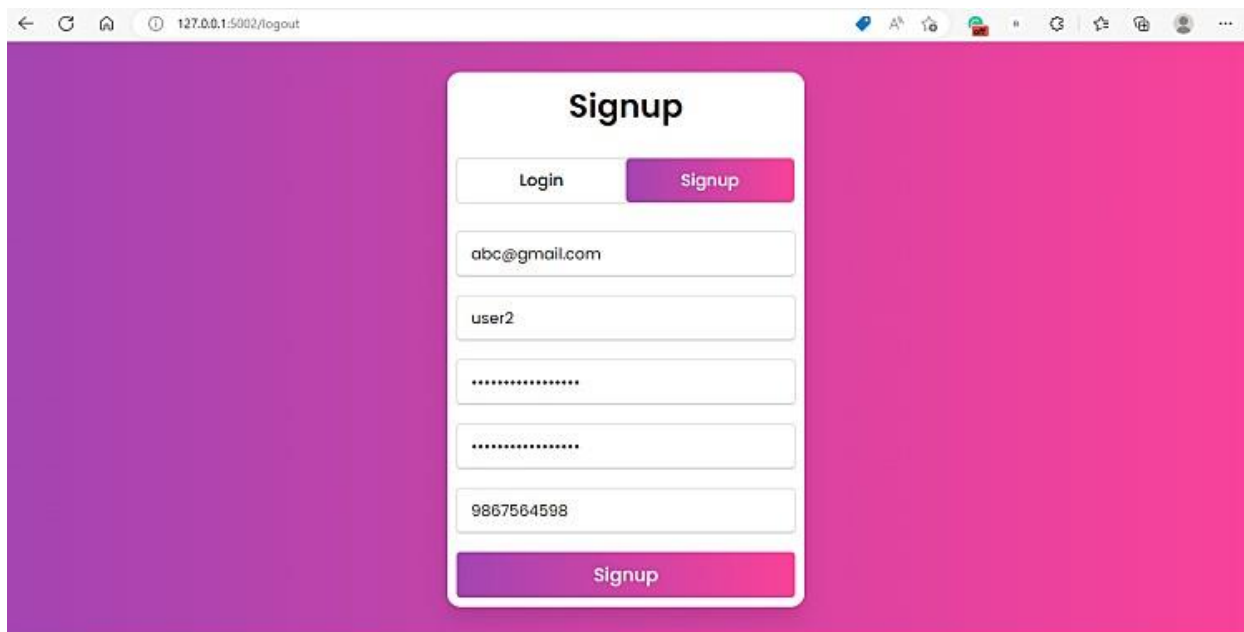
```
<form action="" class="login" id="loginForm">
<div class="field">
<input
type="text"
id="log_in_email"
placeholder="Email"
required
/>
</div>
<div class="field">
<input
type="password"
id="log_in_password"
placeholder="Password"
required
/>
</div>
<div class="pass-link">
<a href="#">Forgot password?</a>
</div>
<div class="field btn">
<div class="btn-layer"></div>
<input type="submit" value="login" id="loginbutton" />
</div>
<div class="signup-link">
Not a member? <a href="">Signup now</a>
</div>
</form>
<form action="" class="signup" id="registerForm">
<div class="field">
<input type="email" placeholder="Email Address" required id="email" name="email" />
</div>
<div class="field">
<input type="text" placeholder="Username" required id="username" name="username" />
</div>
<div class="field">
<input type="password" placeholder="Password" required id="password" name="password"/>
</div>
<div class="field">
```



```

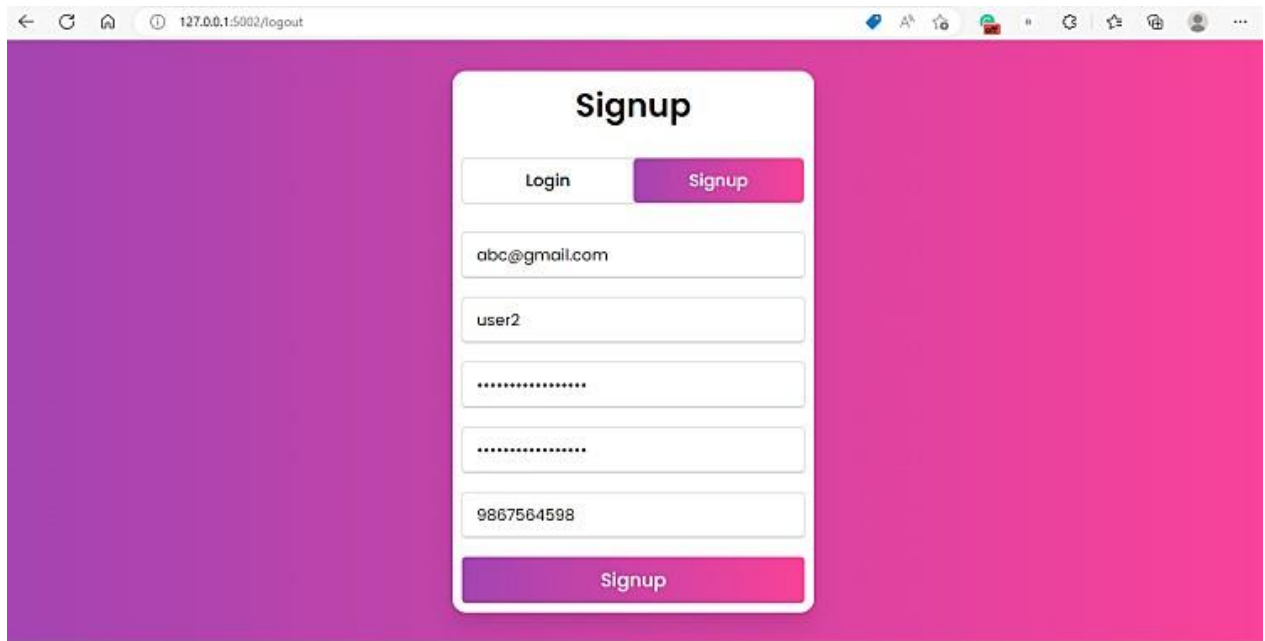
<input type="password" placeholder="Confirm password" required id="confirmPassword"
name="confirmPassword"/>
</div>
<div class="field">
<input type="number" placeholder="Phone Number" required id="phoneNumber" name="phoneNumber"
value=""/>
</div>
<div class="field btn">
<div class="btn-layer"></div>
<input type="submit" value="Signup" id="registerButton" />
</div>
</form>
</div>
</div>
</div>
<span style="" id="message" class=""><p id="message_p"></p></span>
<link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">
<script src="https://code.jquery.com/jquery-3.6.1.js" integrity="sha256-
3zLB5s2uwoUzrXK3BT7AX3FyvojsraNFxCc2vC/7pNI=" crossorigin=
<script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.0/jquery.validate.min.js"> </script>
<script src="{ {url_for('static',filename='index.script.js') } } "></script>
</body>
</html>

```



8. TESTING

Test cases and user acceptance testing

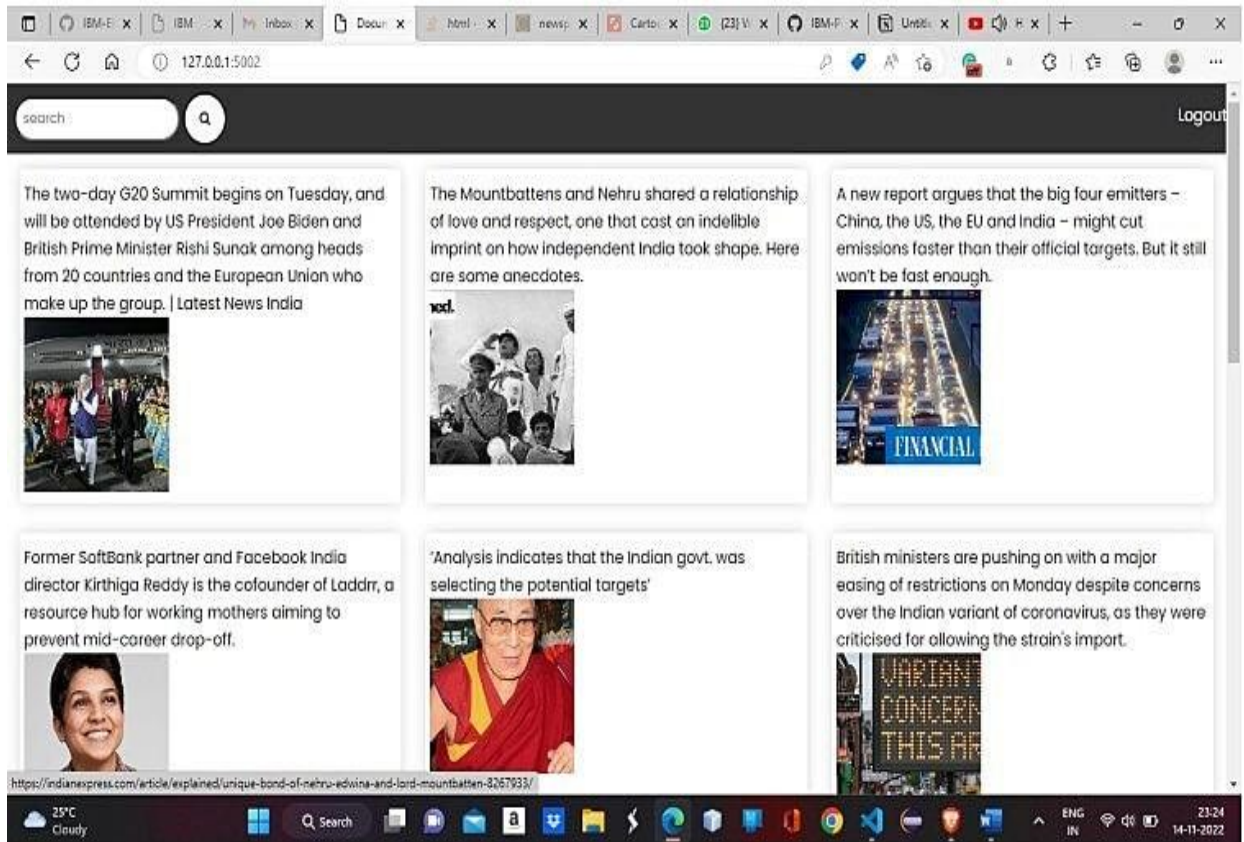


A screenshot of a web browser displaying a Signup form. The browser's address bar shows the URL "127.0.0.1:5002/logout". The form is titled "Signup" and is set against a pink gradient background. It features two buttons at the top: "Login" and "Signup". Below these are five input fields: an email field containing "abc@gmail.com", a username field containing "user2", two password fields (both masked with dots), and a phone number field containing "9867564598". A large "Signup" button is positioned at the bottom of the form.



A screenshot of a web browser displaying a Login form. The form is titled "Login" and is set against a pink gradient background. It features two buttons at the top: "Login" and "Signup". Below these are two input fields: an email field containing "user@gmail.com" and a password field masked with dots. A link labeled "Forgot password?" is located below the password field. A large "login" button is positioned below the input fields. Below the login button, there is a link that says "Not a member? Signup now". At the bottom of the form, a green button displays the text "Login Success".

Performance Metrics



ADVANTAGES OF NEWS TRACKER APPLICATION

1. An online newspaper allows the reader to interact with the paper itself. Readers can now leave comments, watch videos, view photo slideshows and oftentimes contribute their own opinions and written pieces to the paper.
2. Sometimes online editions will even leave out content from the print newspaper's regular edition.
3. The wide range of access points also contributes to the success of the online newspaper. Viewers can get their news straight off their smartphone or tablet computer.
4. News is at their fingertips in an instant.

DISADVANTAGES OF NEWS TRACKER APPLICATION

1. Require data/wifi to get online
2. Companies not making as much money due to free reading for audiences
3. News spreads quicker online - people find out news before they should
4. Lose money - can't get people to pay for digital
5. Older audiences may not access digital platforms
6. Costly to maintain
7. Errors stay online FOREVER

This is fast evolving world with modernization taking a toll on the world, people nowadays prefer to spend time more on the gadgets rather than reading newspaper.

News site is an web application which makes news paper reading much interesting and easier.

The easily usable API and other softwares makes this application very user friendly and most of all save time.

12. FUTURE SCOPE

Implement automated location functionality. This means that as a user moves from one city to another, the local news will change offline browsing can be improved to complete articles more efficiently. This version of app is in the seeding stage and there is a lot of room for future enhancements. Features that can be added for later versions are as follows :

1. Save video
2. Watch this later.

13.APPENDIX

Source code

ajaxRequests.js

```
// const { readFileSync, promises: fsPromises } = require("fs");

$(document).ready(function () {
  testFunction();
  function testFunction(){
    // search_val = $("#searchBox").val();
    search_val="india"
    var queryString = { q: search_val };
    $.ajax({
      method: "POST",
      url: "/search",
      contentType: "application/json;charset=utf-8",
      cache: false,
      datatype: "text",
      data: JSON.stringify(queryString),
      success: function (data) {
        // alert("yes");
        wholeJson = JSON.parse(data);
        articles=wholeJson['articles'];
        // $("#news").text(data)
        showNews(articles);
      },
      error: function (data) {
        // alert("error");
      },
    });
  }
  function showNews(articles) {
    // articles = wholeJson["articles"];
    var newslst = document.getElementById("newsList");
    newslst.innerHTML = "";
    for (var i = 0; i < articles.length; i++) {
      description = articles[i].description;
      image_url = articles[i].urlToImage;
      link_url = articles[i].url;
      var a=document.createElement("a");
      var li = document.createElement("li");
```

```

var image = document.createElement("img");
image.src = image_url;
image.innerHTML = "';
li.innerHTML =
    '<div class="newsLi" id="news_' + i + "'>' + description + "</div>";
li.appendChild(image);
a.setAttribute('href',link_url)
a.setAttribute('target','_blank')
a.appendChild(li);
newslst.appendChild(a);
}
}

```

```

$("#searchButton").click(function () {
    search_val = $("#searchBox").val();
    var queryString = { q: search_val };
    $.ajax({
        method: "POST",
        url: "/search",
        contentType: "application/json;charset=utf-8",
        cache: false,
        datatype: "text",
        data: JSON.stringify(queryString),
        success: function (data) {
            // alert("yes");
            wholeJson = JSON.parse(data);
            articles=wholeJson['articles'];
            // $("#news").text(data)
            showNews(articles);
        },
        error: function (data) {
            alert("error");
        },
    });
});
});
});

```


home.style.css

```
@import url("https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap");
```

```
* {  
  margin: 0;  
  box-sizing: border-box;  
  padding: 0;  
  font-family: "Poppins", sans-serif;  
}
```

```
nav.navbar {  
  position: sticky;  
  top: 0;  
  width: 100vw;  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  background-color: #333;  
}
```

```
nav.navbar .searchContainer {  
  display: inline-block;  
  padding: 0.5rem;
```

```
  /* height:100px; */  
}
```

```
nav.navbar .searchContainer input {  
  padding: 0.5rem;  
  border-radius: 50px;  
}
```

```
nav.navbar .searchContainer button {  
  border-radius: 1.5rem;  
  width: 3rem;  
  height: 3rem;  
  color: #000;  
  background:#fff;  
  cursor: pointer;  
  transition: background 0.5s ease;  
}
```

```
nav.navbar .searchContainer button:hover{  
  background: -webkit-linear-gradient(right, #a445b2, #fa4299, #a445b2, #fa4299);
```

```
    color: #fff;
}
```

```
nav.navbar .userProfile .logout {
    float: right;
    padding: 1rem;
}
```

```
nav.navbar .userProfile .logout a {
    text-decoration: none;
    color: #fff;
}
```

```
.news #newsList {
    list-style: none;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    /* flex-direction: column; */
}
```

```
.news #newsList a {
    border-radius: 5px;
    margin: 1vw;
    box-shadow: 0 0 15px -2px rgba(0, 0, 0, 0.2);
    text-decoration: none;
    color: #000000;
}
```

```
.news #newsList a li {
    padding: 10px 5px;
    overflow: scroll;
}
.news #newsList a li::-webkit-scrollbar {
    display: none;
}
```

```
.news #newsList a li img {
    width: 10rem;
    height: 10rem;
    display: block;
    object-fit: cover;
}
```

index.script.js

```
const loginText = document.querySelector(".title-text .login");
const login = document.querySelector("form.login");
const loginBtn = document.querySelector("label.login");
const signupBtn = document.querySelector("label.signup");
const signupLink = document.querySelector("form .signup-link a");
signupBtn.onclick = () => {
  login.style.marginLeft = "-50%";
  loginText.style.marginLeft = "-50%";
};
loginBtn.onclick = () => {
  login.style.marginLeft = "0%";
  loginText.style.marginLeft = "0%";
};
signupLink.onclick = () => {
  signupBtn.click();
  return false;
};
```

// Above code is completely for stylings

```
$(document).ready(function () {
  function handleError(message) {
    $("#message").addClass("errorForMessage", 500, callBack);
    $("#message").text(message);
    $("#message").animate(
      {
        height: "60px",
        padding: "10px",
      },
      500
    );
    function callBack() {
      setTimeout(function () {
        $("#message").animate(
          {
            height: "0",
            padding: "0",
          },
          500
        );
      });
    }
  }
});
```

```

    setTimeout(function () {
        $("#message").removeClass("errorForMessage");
    }, 1500);
}, 1500);
}
}

```

```

function handleSuccess(message) {
    return new Promise(function (resolve, reject) {
        $("#message").addClass("successForMessage", 500, callBack);
        $("#message").text(message);
        $("#message").animate(
            {
                height: "60px",
                padding: "10px",
            },
            500
        );
        function callBack() {
            setTimeout(function () {
                $("#message").animate(
                    {
                        height: "0",
                        padding: "0",
                    },
                    500
                );
                setTimeout(function () {
                    $("#message").removeClass("successForMessage");
                    resolve();
                }, 1500);
            }, 1500);
        }
    });
}

```

```

function postForm(path, params, method) {
    method = method || "post";

    var form = document.createElement("form");
    form.setAttribute("method", method);
    form.setAttribute("action", path);

```

```

    for (var key in params) {
        if (params.hasOwnProperty(key)) {
            var hiddenField = document.createElement("input");
            hiddenField.setAttribute("type", "hidden");
            hiddenField.setAttribute("name", key);
            hiddenField.setAttribute("value", params[key]);

            form.appendChild(hiddenField);
        }
    }

    document.body.appendChild(form);
    form.submit();
}

```

```

$("#loginForm").submit(function (e) {
    e.preventDefault();
    email = $("#log_in_email").val();
    password = $("#log_in_password").val();
    var data = { email: email, password: password };
    $.ajax({
        method: "POST",
        url: "/loginvalidate",
        contentType: "application/json;charset=utf-8",
        cache: false,
        datatype: "text",
        data: JSON.stringify(data),
        success: function (data) {
            if (data == "success") {
                handleSuccess("Login Success").then(function(){
                    postForm("/login",{username:username})
                });
            } else {
                handleError("Invalid credentials");
            }
        },
        error: function (data) {
            alert("error" + data);
        },
    });
    return false;
});

```

// Above code is for login

```
function checkForExistingUser(data) {  
  return new Promise(function (resolve, reject) {  
    var emailJson = { email: data };  
    $.ajax({  
      method: "POST",  
      url: "/checkForExistingUser",  
      contentType: "application/json",  
      cache: false,  
      datatype: "text",  
      async: false,  
      data: JSON.stringify(emailJson),  
      success: function (data_from_server) {  
        if (data_from_server !== "true") {  
          console.log("yes");  
          resolve(data_from_server);  
        } else {  
          handleError("Email Id is already in use");  
          reject(data_from_server);  
        }  
      },  
      error: function (data_from_server) {  
        reject(data_from_server);  
      },  
    });  
  });  
}
```

```
$.validator.addMethod(  
  "validatePassword",  
  function (value, element) {  
    return (  
      this.optional(element) ||  
      /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{8,}$/.test(  
        value  
      )  
    );  
  },  
  "Password must contain 1 uppercase, 1 lowercase, 1 number, 1 special character"  
);
```

```
$.validator.addMethod(  
  "validatePhone",
```

```
function (value, element) {  
    return this.optional(element) || /^[6-9]\d{9}$/.test(value);  
},  
"Enter valid Phone Number"  
);
```

```
$("#registerForm").validate({  
    rules: {  
        email: {  
            required: true,  
            email: true,  
        },  
        username: {  
            required: true,  
        },  
        password: {  
            required: true,  
            validatePassword: true,  
        },  
        confirmPassword: {  
            required: true,  
            equalTo: "#password",  
        },  
        phoneNumber: {  
            required: true,  
            validatePhone: true,  
        },  
    },  
    messages: {  
        email: {  
            required: "Email is required",  
            email: "Enter valid email",  
        },  
        username: {  
            required: "Username is required",  
        },  
        password: {  
            required: "Password is required",  
            minlength: "Password must contains minimum 8 characters",  
        },  
        confirmPassword: {  
            equalTo: "Password must be equal to confirm Password",  
            required: "Confirm Password is required",  
        },  
    },  
});
```

```

    },
    phoneNumber: {
        required: "Phone Number is required",
    },
},
submitHandler: function (form) {
    email = $("#email").val();
    checkForExistingUser(email)
    .then(function () {
        console.log("mail is available");
        mail = $("#email").val();
        username = $("#username").val();
        password = $("#password").val();
        phone = $("#phoneNumber").val();
        jsonObject = {
            mail: mail,
            username: username,
            password: password,
            phone: phone,
        };
        $.ajax({
            method: "POST",
            url: "/register",
            contentType: "application/json",
            cache: false,
            datatype: "text",
            async: false,
            data: JSON.stringify(jsonObject),
            success: function (data_from_server) {
                if (data_from_server == "success") {
                    // console.log("yes submitted");
                    handleSuccess("Sign up Success!").then(function(){
                        window.location.href = "/";
                    })
                } else {
                    handleError("Sign up Failed");
                    // console.log("not submitted");
                }
            },
            error: function (data_from_server) {
                console.log("not submitted");
            },
        });
    });
}

```



```

    })
    .catch(function () {
        console.log("mail is already used");
    });
    return false;
},
});
});

```

index.style.css

```

/* * {
    padding: 0;
    box-sizing: border-box;
    margin: 0;
}
.container {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 50vw;
    text-align: center;
}
.container form input[type=text],input[type=password] {
    margin: 5px;
    border: 1px solid #000000;
    border-radius: 50px;
} */
/*

```

```

=====
===== */

```

```

@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
html,body{
    overflow: hidden;

```

```
display: grid;
height: 100%;
width: 100%;
place-items: center;
background: -webkit-linear-gradient(left, #a445b2, #fa4299);
}
::selection{
background: #fa4242;
color: #fff;
}
.wrapper{
/* float: right; */
overflow: hidden;
max-width: 390px;
background: #fff;
padding: 10px;
border-radius: 15px;
box-shadow: 0px 15px 20px rgba(0,0,0,0.1);
margin: 2rem;
}
.wrapper .title-text{
display: flex;
width: 200%;
}
.wrapper .title{
width: 50%;
font-size: 35px;
font-weight: 600;
text-align: center;
transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);
}
.wrapper .slide-controls{
position: relative;
display: flex;
height: 50px;
width: 100%;
overflow: hidden;
margin: 30px 0 10px 0;
justify-content: space-between;
border: 1px solid lightgrey;
border-radius: 5px;
}
.slide-controls .slide{
```

```
height: 100%;
width: 100%;
color: #fff;
font-size: 18px;
font-weight: 500;
text-align: center;
line-height: 48px;
cursor: pointer;
z-index: 1;
transition: all 0.6s ease;
}
.slide-controls label.signup{
color: #000;
}
.slide-controls .slider-tab{
position: absolute;
height: 100%;
width: 50%;
left: 0;
z-index: 0;
border-radius: 5px;
background: -webkit-linear-gradient(left, #a445b2, #fa4299);
transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);
}
input[type="radio"]{
display: none;
}
#signup:checked ~ .slider-tab{
left: 50%;
}
#signup:checked ~ label.signup{
color: #fff;
cursor: default;
user-select: none;
}
#signup:checked ~ label.login{
color: #000;
}
#login:checked ~ label.signup{
color: #000;
}
#login:checked ~ label.login{
cursor: default;
```

```
    user-select: none;
}
.wrapper .form-container{
    width: 100%;
    overflow: hidden;
}
.form-container .form-inner{
    display: flex;
    width: 200%;
}
.form-container .form-inner form{
    width: 50%;
    transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);
}
.form-inner form .field{
    height: 50px;
    width: 100%;
    margin-top: 20px;
}
.form-inner form .field input{
    height: 100%;
    width: 100%;
    outline: none;
    padding-left: 15px;
    border-radius: 5px;
    border: 1px solid lightgrey;
    border-bottom-width: 2px;
    font-size: 17px;
    transition: all 0.3s ease;
}
.form-inner form .field input:focus{
    border-color: #fc83bb;
    /* box-shadow: inset 0 0 3px #fb6aae; */
}
.form-inner form .field input::placeholder{
    color: #999;
    transition: all 0.3s ease;
}
form .field input:focus::placeholder{
    color: #b3b3b3;
}
.form-inner form .pass-link{
    margin-top: 5px;
```

```

}
.form-inner form .signup-link{
  text-align: center;
  margin-top: 30px;
}
.form-inner form .pass-link a,
.form-inner form .signup-link a{
  color: #fa4299;
  text-decoration: none;
}
.form-inner form .pass-link a:hover,
.form-inner form .signup-link a:hover{
  text-decoration: underline;
}
form .btn{
  height: 50px;
  width: 100%;
  border-radius: 5px;
  position: relative;
  overflow: hidden;
}
form .btn .btn-layer{
  height: 100%;
  width: 300%;
  position: absolute;
  left: -100%;
  background: -webkit-linear-gradient(right, #a445b2, #fa4299, #a445b2, #fa4299);
  border-radius: 5px;
  transition: all 0.4s ease;;
}
form .btn:hover .btn-layer{
  left: 0;
}
form .btn input[type="submit"]{
  height: 100%;
  width: 100%;
  z-index: 1;
  position: relative;
  background: none;
  border: none;
  color: #fff;
  padding-left: 0;
  border-radius: 5px;

```

```
font-size: 20px;  
font-weight: 500;  
cursor: pointer;  
}
```

```
#message{  
  visibility: hidden;  
  padding: 10px;  
  position: absolute;  
  top: 80%;  
  height: 0;  
  left: 40%;  
  font-size: 4vh;  
  overflow: hidden;  
  border-radius: 5px;  
}
```

```
.errorForMessage{  
  visibility: visible !important;  
  background-color: rgb(200, 0, 0,.5);  
  color: #ffffff;  
}
```

```
.successForMessage{  
  visibility: visible !important;  
  background-color: rgb(46, 226, 46);  
  color: #ffffff;  
}
```

templates

home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <script
      src="https://code.jquery.com/jquery-3.6.1.js"
      integrity="sha256-3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFx Cc2vC/7pNI="
      crossorigin="anonymous"
    ></script>
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
    />
    <link rel="stylesheet"
      href="{{url_for('static',filename='home.style.css')}}"
    />
  </head>
  <body>
    <nav class="navbar">
      <div class="searchContainer">
        <input
          type="text"
          name="searchBox"
          id="searchBox"
          placeholder="search"
        />
        <button id="searchButton">
          <i class="fa fa-search" aria-hidden="true"></i>
        </button>
      </div>
      <div class="userProfile">
        <div class="logout">
          <a href="{{url_for('logout')}}" target="" rel="">Logout</a>
        </div>
      </div>
    </body>
```

```

</nav>
<div class="news">
  <ul id="newsList"></ul>
</div>
<script src="{{url_for('static',filename='ajaxRequests.js')}}"></script>
</body>
</html>

```

```
{#
```

```

<h1>hi u have succefully logged in</h1>
<a href="{{url_for('logout')}}">Logout</a>
<input type="text" placeholder="search" id="search" />
<script
  src="https://code.jquery.com/jquery-3.6.1.js"
  integrity="sha256-3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFx Cc2vC/7pNI="
  crossorigin="anonymous"
></script>
<input type="button" value="get news !" id="getNews" />
<ul id="newslist"></ul>
<script src="{{url_for('static',filename='ajaxRequests.js')}}"></script>
#}

```

index.html

```

<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8" />
    <title>Login and Registration in HTML | CodingNepal</title>
    <link rel="stylesheet" href="../static/index.style.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  </head>
  <body>
    <div class="wrapper">
      <div class="title-text">
        <div class="title login">Login</div>
        <div class="title signup">Signup</div>
      </div>
      <div class="form-container">
        <div class="slide-controls">
          <input type="radio" name="slide" id="login" checked />

```



```



```

```

    </div>
    <div class="field">
        <input type="password" placeholder="Password" required id="password" name="password"
value="" />
    </div>
    <div class="field">
        <input type="password" placeholder="Confirm password" required id="confirmPassword"
name="confirmPassword" value="" />
    </div>
    <div class="field">
        <input type="number" placeholder="Phone Number" required id="phoneNumber"
name="phoneNumber" value="" />
    </div>
    <div class="field btn">
        <div class="btn-layer"></div>
        <input type="submit" value="Signup" id="registerButton" />
    </div>
</form>
</div>
</div>
</div>
<span style="" id="message" class=""><p id="message_p"></p></span>
<link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">
<script src="https://code.jquery.com/jquery-3.6.1.js" integrity="sha256-
3zlB5s2uwoUzrXK3BT7AX3FyvojsraNFxCc2vC/7pNI=" crossorigin="anonymous"></script>
<script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.0/jquery.validate.min.js">
</script>
<script src="{{url_for('static',filename='index.script.js')}}"></script>
</body>
</html>

```

app.py

```
from flask import Flask, render_template, session, request, redirect, make_response
import ibm_db
from ibm_db2_connect import db2
import news
import json
app = Flask(__name__)
app.secret_key = "secret_key"

@app.route('/', methods=['GET', 'POST'])
def index():
    if session.get('logged_in'):
        return render_template('home.html')
    return render_template('index.html')

@app.route('/login', methods=['POST'])
def login():
    form_user = request.form['username']
    session['logged_in'] = form_user
    return redirect('/')

@app.route('/loginvalidate', methods=['POST', 'GET'])
def loginvalidate():
    conn = db2.get_conn()
    data = request.get_json();
    form_user = data['email']
    form_password = data['password']
    sql = 'SELECT username from user WHERE mail=? and password=?'
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, form_user)
```

```

    ibm_db.bind_param(stmt, 2, form_password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        response = make_response('success',200)
    else:
        response = make_response('failure',200)
    response.mimetype = "text/plain"
    return response

@app.route('/register',methods=['POST'])
def register():
    try:
        conn = db2.get_conn()
        data=request.get_json();
        form_user = data['username']
        form_password = data['password']
        form_mail = data['mail']
        form_phone = data['phone']
        sql = 'insert into user values(?,?,?,?)'
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, form_user)
        ibm_db.bind_param(stmt, 2, form_password)
        ibm_db.bind_param(stmt, 3, form_mail)
        ibm_db.bind_param(stmt, 4, form_phone)
        ibm_db.execute(stmt)
        response = make_response('success',200)
    except Exception as e:
        response = make_response('failure',200)
        print(e)
    return response

@app.route('/checkForExistingUser',methods=['POST'])

```

```

def checkForExistingUser():
    conn = db2.get_conn()
    data=request.get_json();
    email=data['email'];
    print(email)
    sql = 'SELECT mail from user WHERE mail=?'
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        response = make_response('true',200)
    else:
        response = make_response('false',200)
    response.mimetype = "text/plain"
    return response

```

```

@app.route('/search',methods=['POST'])

```

```

def getNews():
    getRequests=request.get_json()
    articles=news.getNews(getRequests)
    # print(json.dumps(articles))
    response = make_response(json.dumps(articles),200)
    response.mimetype="text"
    return response

```

```

@app.route('/logout')

```

```

def logout():
    session.clear()
    return render_template('index.html')

```

```
if __name__ == '__main__':  
    app.run(debug=True, port=5002)
```

config.py

```
DATABASE="<ypur db>"  
HOSTNAME="<your hostname>"  
PORT="<your port>"  
SECURITY="SSL"  
SSLServerCertificate="<your certificate>"  
UID="<your UID>"  
PWD="<your PASSWORD>"  
API_KEY="<your API key>"
```

ibm_db2_connect.py

```
import ibm_db  
import config as c  
class db2:  
      
    connectionString="DATABASE={0};HOSTNAME={1};PORT={2};SECURITY={3};SSLServerCertificate={4};UID={5};PWD={6}".format(c.DATABASE,c.HOSTNAME,c.PORT,c.SECURITY,c.SSLServerCertificate,c.UID,c.PWD)  
    conn = ibm_db.connect(connectionString,"")  
    def get_conn():  
        return db2.conn
```

news.py

```
import requests
```

```
from config import API_KEY
```

```
def getNews(query:dict):
```

```
    query_string="q="+query.get('q')
```

```
    main_url="https://newsapi.org/v2/top-headlines?"
```

```
    final_url="{0}{1}&apiKey={2}".format(main_url,query_string,API_KEY)
```

```
    article=requests.get(final_url)
```

```
    return article.json()
```