HX8001 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

NEWS TRACKER APPLICATION

TEAM ID: PNT2022TMID50246

FACULTY MENTOR NAME: MRS. W. VINOTHINI MARY

TEAM LEADER: P. MUTHUSELVI

TEAM MEMBERS: J. DIVYA BHARATHI
R. BALA PRIYA
A. NIVETHA PAPPA

PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview NewsTracker Application is a fullstack 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.

1.2 Purpose 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

2.1 Existing problem

Physical newspapers are old fashioned in this digital era. They cost money to buy, can easily be damaged, limited amount of information, not flexible to modifications ,poor quality. Sometimes may show unrelevant and updated news.

2.2 References

s.no	Paper title	Author	Published date	Implementation
1	Following the Fed with a News Tracker	Michael William McCracken	January, 2012	The paper is not a technical paper but is essentially a statistical paper on how should one conclude whether the data have come in stronger, weaker or as expected. This is based on the CitiGroup U.S Economic Surprise Index.
2	Topic Detection and Tracking in News Articles	Sagar Patel, Sanket suthar, Sandip Patel, Neha Patel	March, 2015	1. Pre-processing 2. Tokenization 3.Stemming/Lemmization 4. Vector Space Model 5. Topic tracking
3	An End-to-end Weaklysupervised News Aggregation Framework	Xijin Tang, Xiaohui	June, 2022	The framework combines Snorkel based weakly Supervised classification, Latent Dirichlet Allocation (LDA) topic modeling, and topic signal detection model to classify and aggregate unlabeled news texts and ultimately generate visualized results containing news categories, news topics, and temporal topic relationships. This paper uses constructed knowledge thesaurus and the Snorkel method to weakly supervise the classification of unlabeled news with no manual tagging. Subsequently, we utilize LDA to generate the topics and obtain the

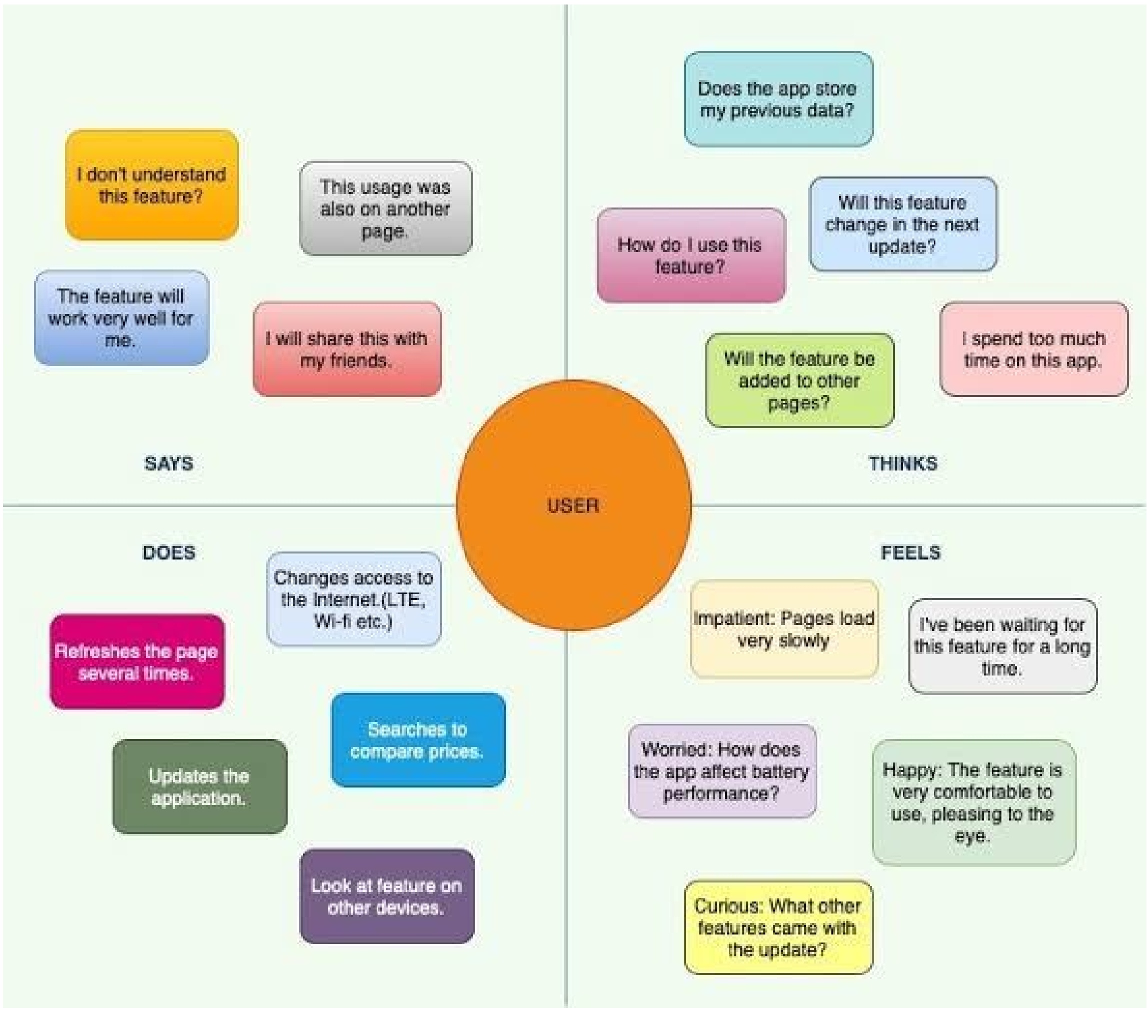
				signal value of each topic based on the topic signal detection function. Finally, we establish the temporal topic relationships and get the visualized results of news aggregation.
4	Exploring mobile news reading interactions for news app personalisation	Marios Constantinides ,John Dowell , David Johson, Sylvain Malacria	August, 2015	1. Identification of news reader types 2. Interaction logging and classification study 3. Deployment and data collection 4. Predicting News reader types 5. Adaptive UI
5	Innovative Application For News Tracker Application	Dr.C.K.Gomathy, Dr.V.Geetha, Peddireddy Abhiram, Marios Constantinides	September, 2020	This paper aimed at developing an online news management system that is of information to either a college. Online news management system provides a simple interface for maintenance of college information. The creation management of accurate,up-to-date information regarding to college.The main objective for developing this project is provide all the functionality related to latest news and it tracks.

2.3 Problem Statement Definition

Newspaper contains limited , non user/reader specific, Location specific news . There are multiple news-sharing apps available which can be used by a single user and are often spammed with notifications. There is also a lot of unwanted news which gets shared. So it may take a lot of time for the user to find the news he/she likes. 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

3. IDEATION & PROPOSED SOLUTION

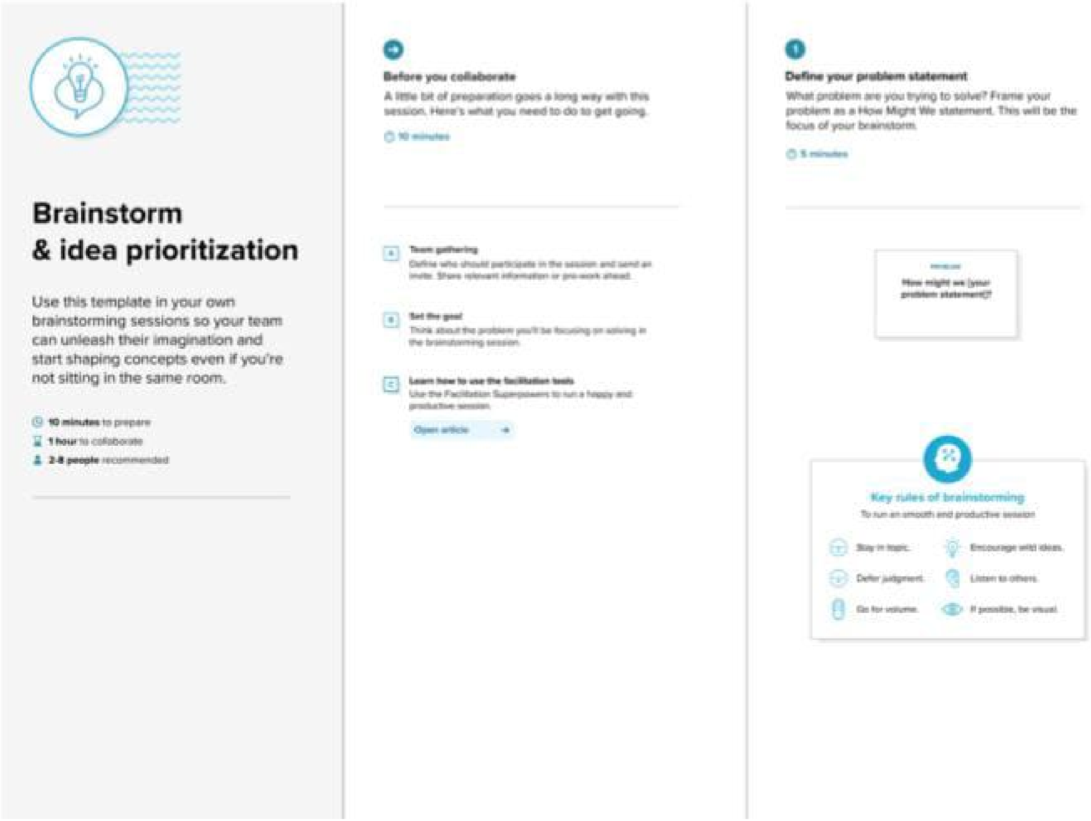
3.1 Empathy map canvas


I don't understand this feature?
This usage was also on another page.
The feature will work very well for me.
I will share this with my friends.
SAYS
Does the app store my previous data?
Will this feature change in the next update?
How do I use this feature?
I spend too much time on this app.
Will the feature be added to other pages?
THINKS
USER
DOES
Changes access to the Internet.(LTE, Wi-fi etc.)
Refreshes the page several times.
Searches to compare prices.
Updates the application.
Look at feature on other devices.
FEELS
Impatient: Pages load very slowly
I've been waiting for this feature for a long time.
Worried: How does the app affect battery performance?
Happy: The feature is very comfortable to use, pleasing to the eye.
Curious: What other features came with the update?

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



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
- **2-8 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 →

1 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

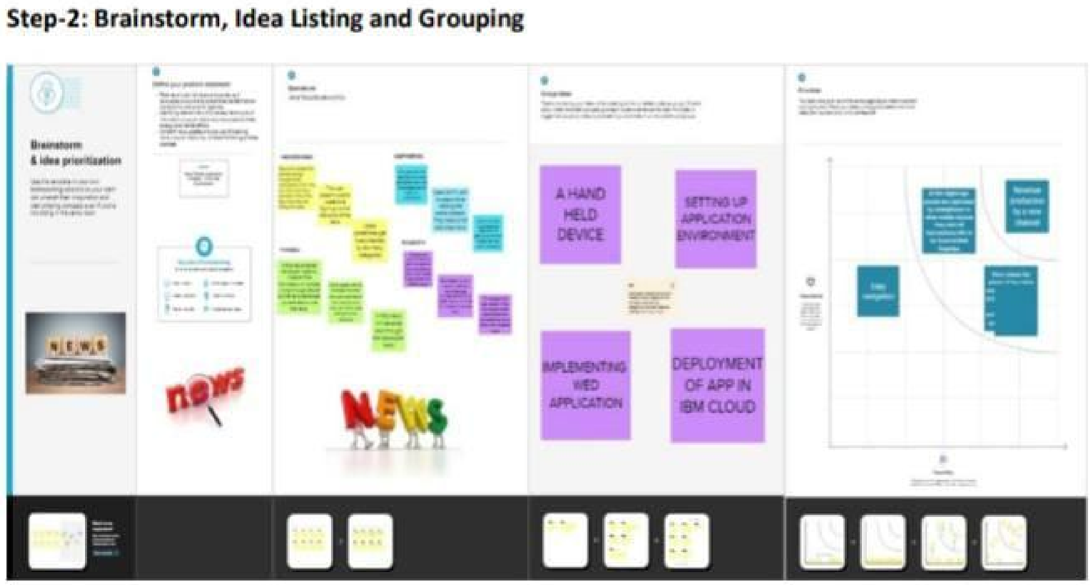
[illegible]
How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

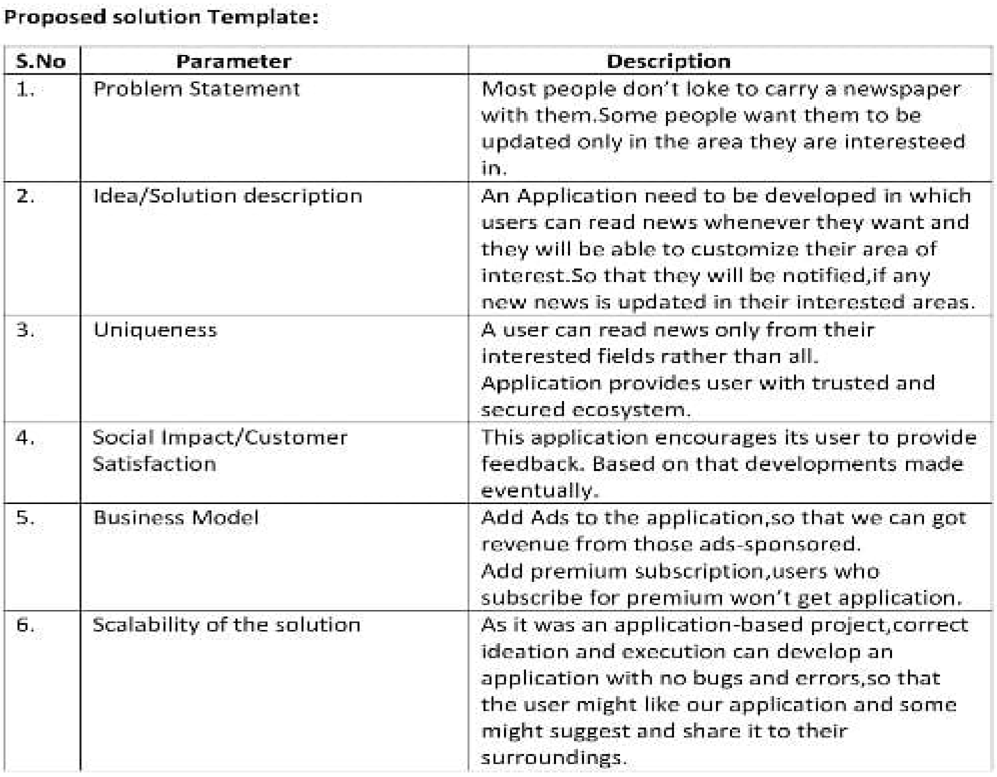
Step-2: Brainstorm, Idea Listing and Grouping



Step-2: Brainstorm, Idea Listing and Grouping

Brainstorm
& idea prioritization
NEWS
news
NEWS
A HAND HELD DEVICE
SETTING UP APPLICATION ENVIRONMENT
IMPLEMENTING WEB APPLICATION
DEPLOYMENT OF APP IN IBM CLOUD

3.3 Proposed solution

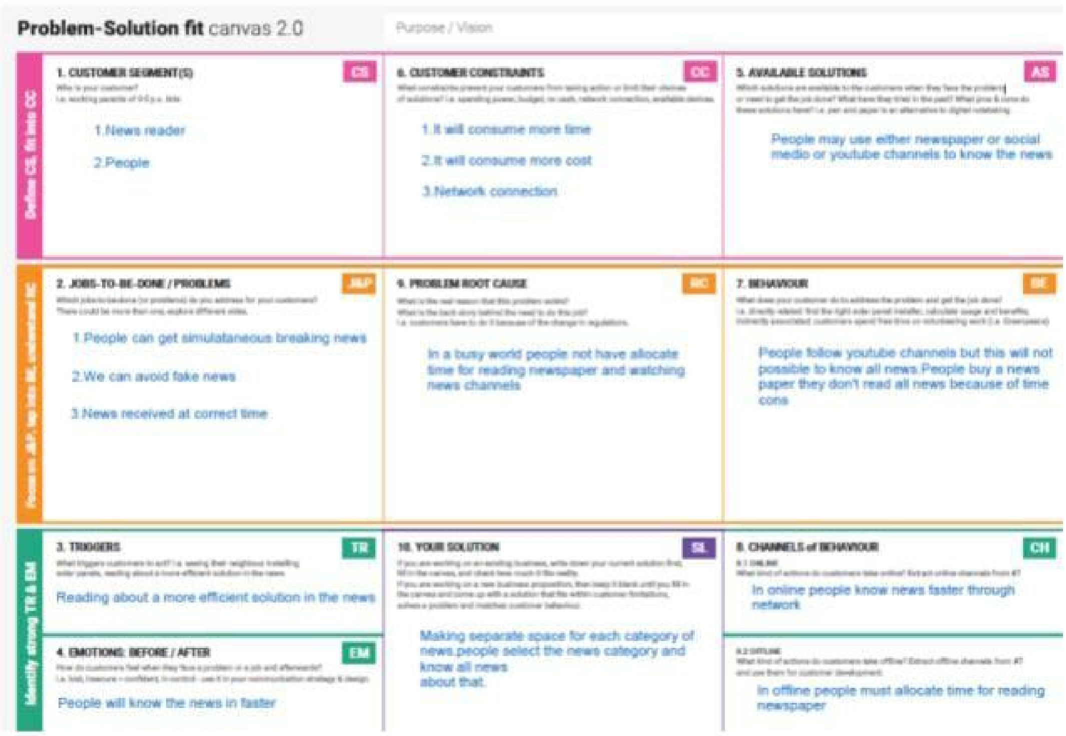


Proposed solution Template:

S.No	Parameter	Description
1.	Problem Statement	Most people don't loke to carry a newspaper with them.Some people want them to be updated only in the area they are interesteed in.
2.	Idea/Solution description	An Application need 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.	Uniqueness	A user can read news only from their interested fields rather than all. Application provides user with trusted and secured ecosystem.
4.	Social Impact/Customer Satisfaction	This application encourages its user to provide feedback. Based on that developments made eventually.
5.	Business Model	Add Ads to the application,so that we can got revenue from those ads-sponsored. Add premium subscription,users who subscribe for premium won't get application.
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.

3.4 Problem Solution

Problem solution:



Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S) CS

1.News reader

2.People

6. CUSTOMER CONSTRAINTS CC

1.It will consume more time

2.It will consume more cost

3.Network connection

5. AVAILABLE SOLUTIONS AS

People may use either newspaper or social medio or youtube channels to know the news

2. JOBS-TO-BE-DONE / PROBLEMS J&P

1.People can get simulataneous breaking news

2.We can avoid fake news

3.News received at correct time

9. PROBLEM ROOT CAUSE RC

In a busy world people not have allocate time for reading newspaper and watching news channels

7. BEHAVIOUR BE

People follow youtube channels but this will not posible to know all news.People buy a news paper they don't read all news because of time cons

3. TRIGGERS TR

Reading about a more efficient solution in the news

4. EMOTIONS: BEFORE / AFTER EM

People will know the news in faster

10. YOUR SOLUTION SL

Making separate space for each category of news.people select the news category and know all news
about that.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

In online people know news faster through network

8.2 OFFLINE

In offline people must allocate time for reading newspaper

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic) S	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration through Email
2	User Confirmation	Confirmation via Email using Sendgrid
3	User Preffered Topics	Collecting user’s favourite topics
4	Collecting News	Collecting news from rapid API at regular Intervals
5	Bookmark	Bookmarking Favourite news for the user

6	Login	Loging in the user using the registered email and password
7	News feed	Showing the news fetched from the API to the user

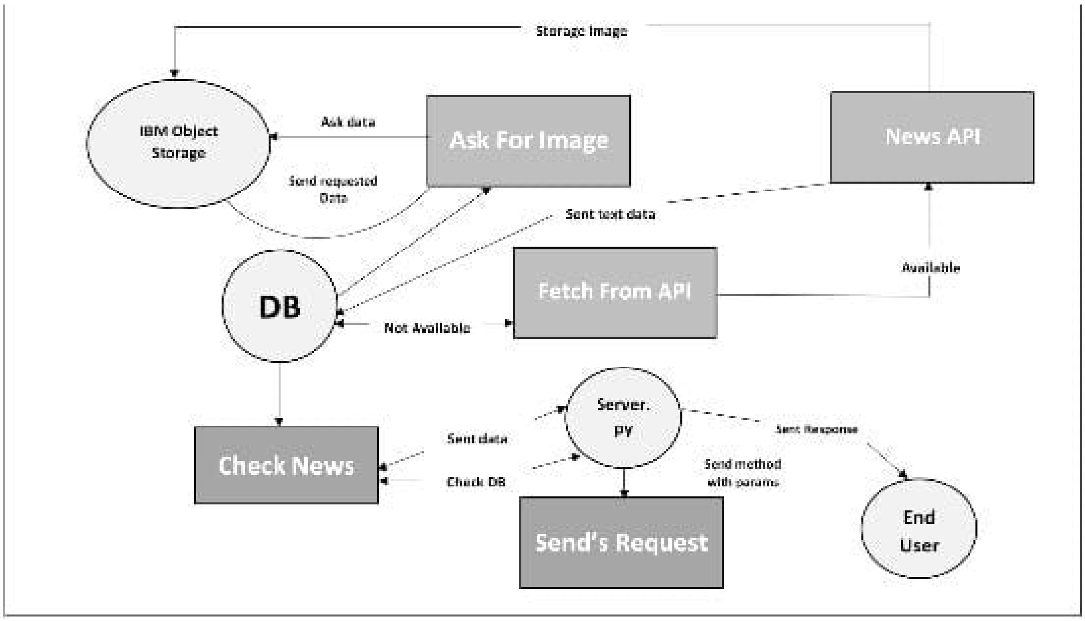
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
1	Usability	The app should be able to used by all people
2	Security	The app should authenticate the legitimate users and should restrict bot attacks
3	Reliability	The app should show the news which are non fake and spam free
4	Performance	The app should be accessible in older devices too
5	Availability	The app should be available to all devices
6	Scalability	The app should handle multiple users and be designed in such a way for future upgrade

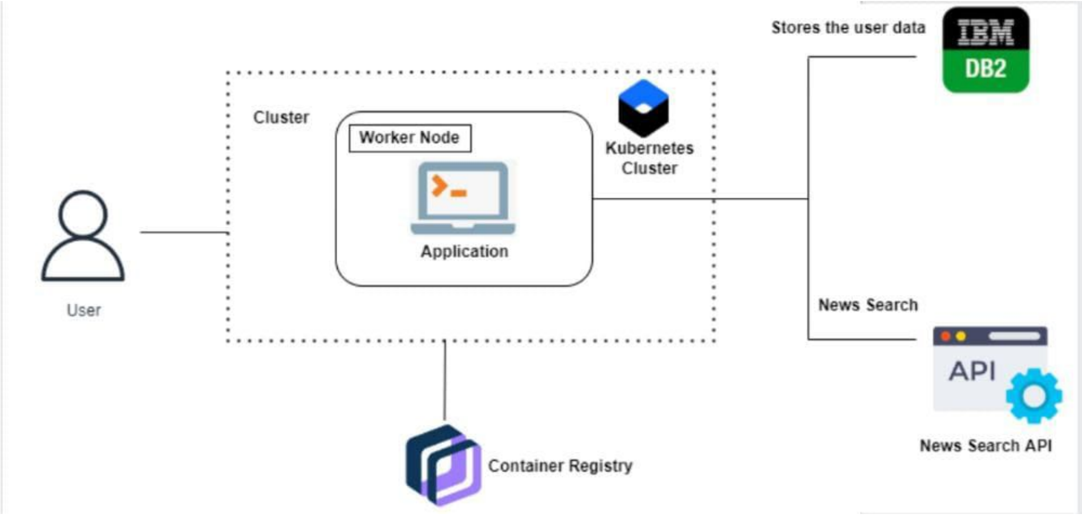
5. PROJECT DESIGN

5.1 Data Flow Diagrams

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



5.2 Solution & Technical Architecture

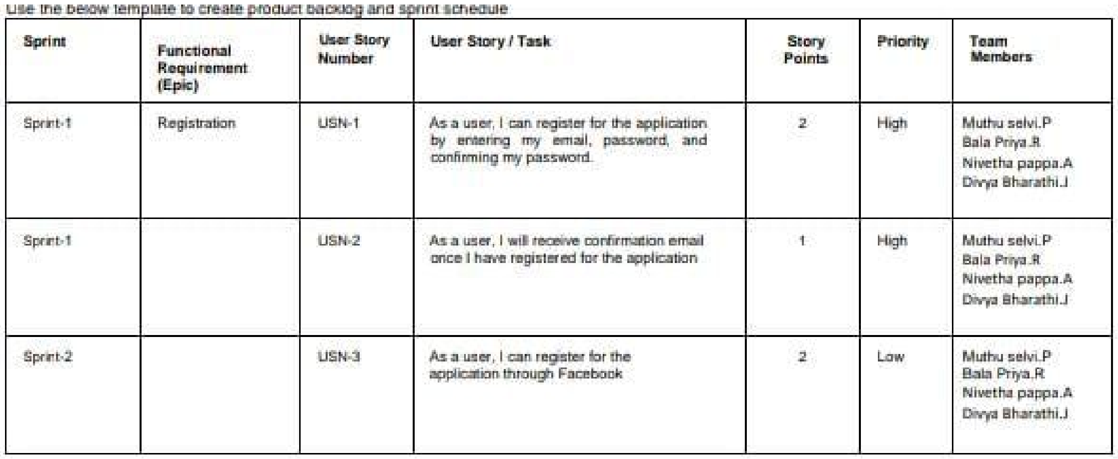


5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User story	Acceptance criteria	Priority	Release
Normal user	Registration	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
	Email Verification	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
	Favourite topics	3	As a user, I can choose my favourite topics	I can see all my preferred news under for you category	Medium	Sprint-2
	Login	4	As a user, I can login	I can access to all the news	High	Sprint-1

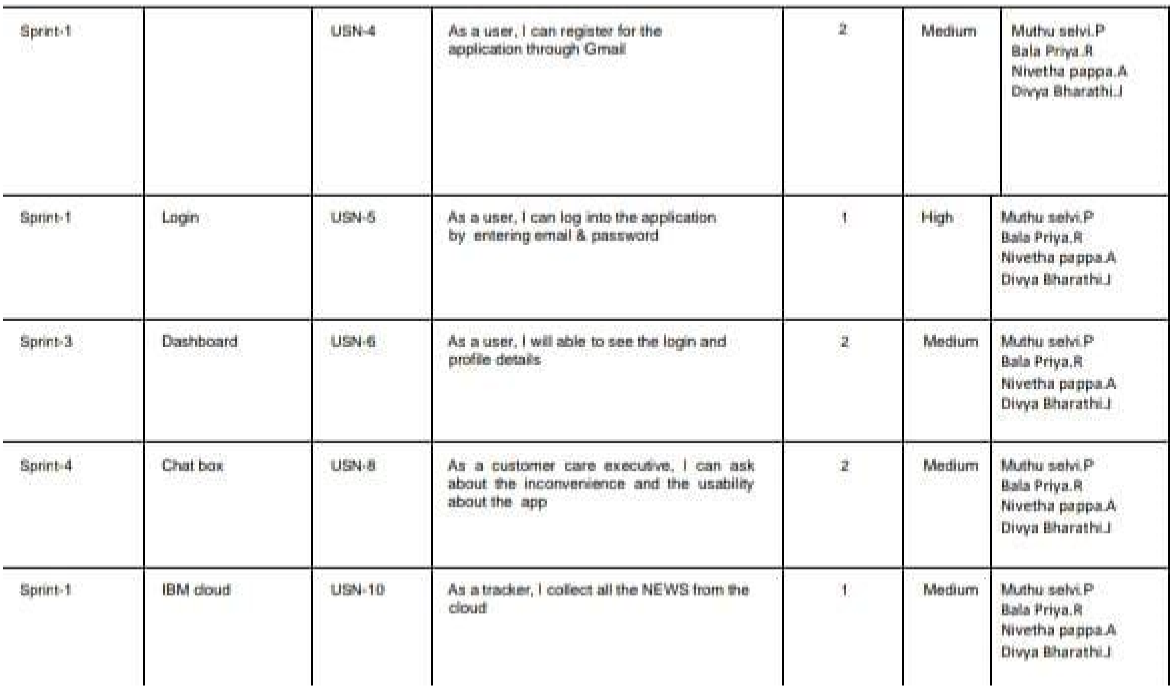
			with the email and password into the app			
	Dashboard	5	As a user, I can see all the news under specific tab	I can view all the news	Mediu m	Sprint-3
	Bookmark	6	As a user, I can bookmark my favourite news topics	I can later view my bookmarked news	Low	Sprint-2
	Access	7	As a user, I can access the site anywhere and everywhere	I can view the site by typing the URL of the site	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

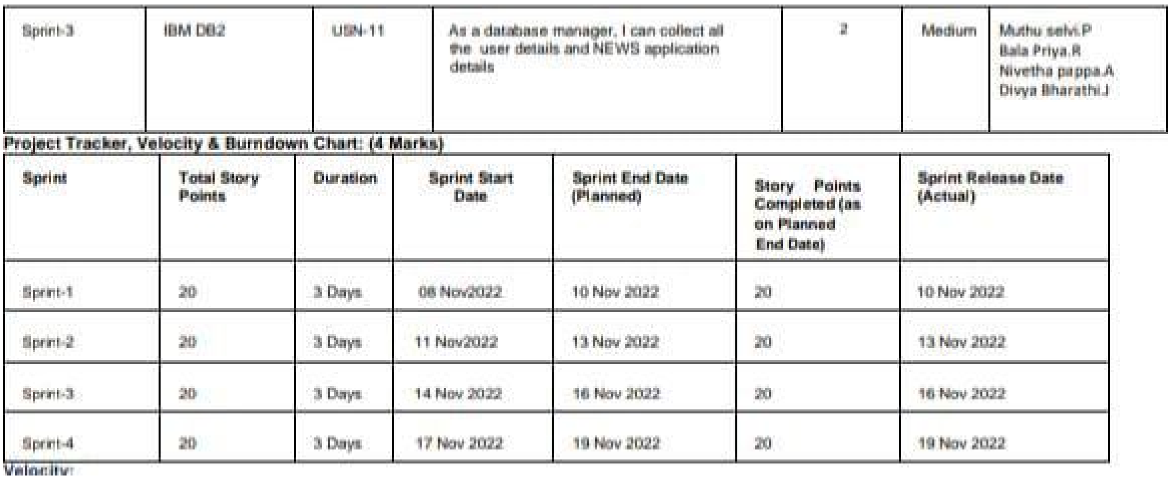


Use the below template to create product backlog and sprint schedule

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	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J



Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
Sprint-3	Dashboard	USN-6	As a user, I will able to see the login and profile details	2	Medium	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
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	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J
Sprint-1	IBM cloud	USN-10	As a tracker, I collect all the NEWS from the cloud	1	Medium	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J



Sprint-3	IBM DB2	USN-11	As a database manager, I can collect all the user details and NEWS application details	2	Medium	Muthu selvi.P Bala Priya.R Nivetha pappa.A Divya Bharathi.J

Project Tracker, Velocity & Burndown Chart: (4 Marks)

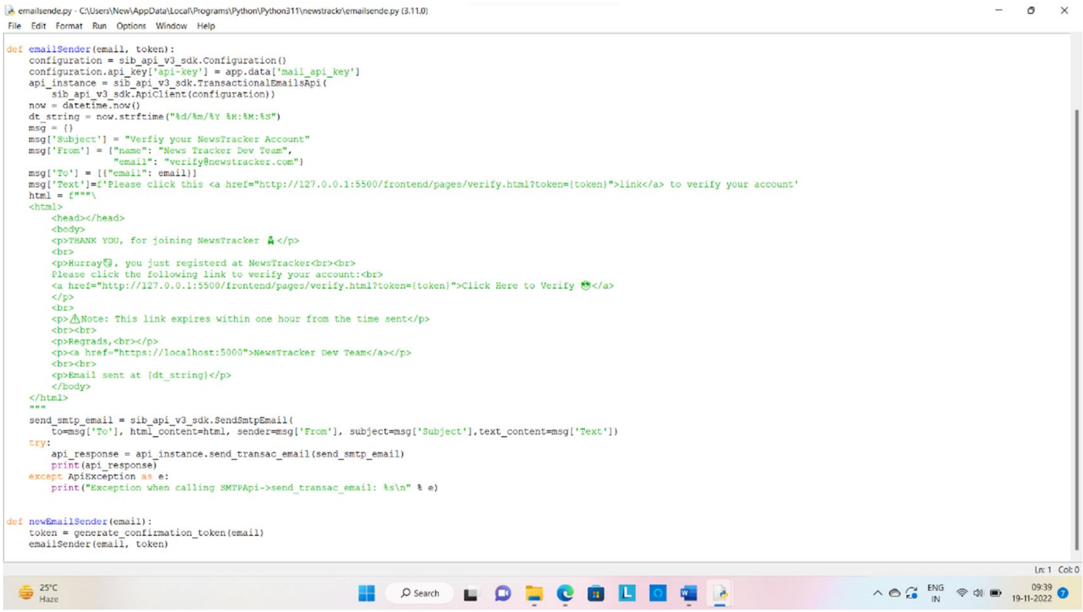
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	3 Days	08 Nov2022	10 Nov 2022	20	10 Nov 2022
Sprint-2	20	3 Days	11 Nov2022	13 Nov 2022	20	13 Nov 2022
Sprint-3	20	3 Days	14 Nov 2022	16 Nov 2022	20	16 Nov 2022
Sprint-4	20	3 Days	17 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

7.1 Feature 1

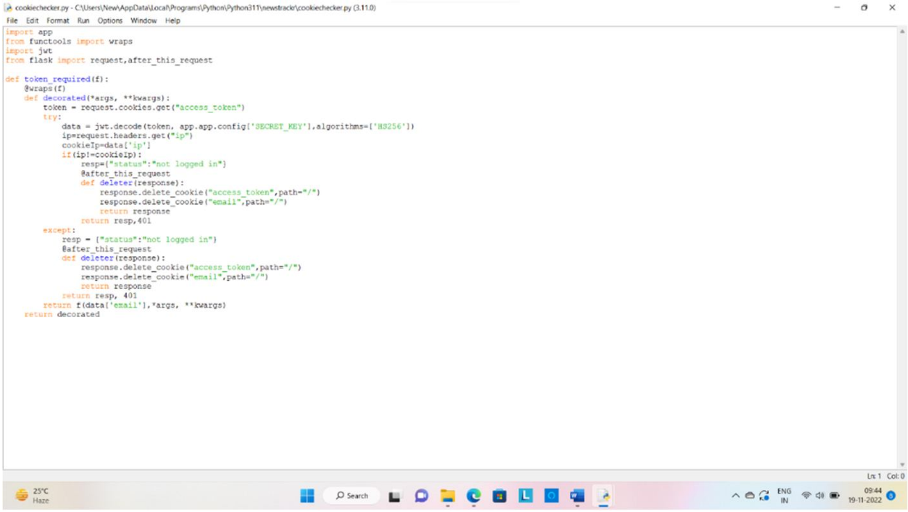
Verification mail sender


emailsende.py - C:\Users\New\AppData\Local\Programs\Python\Python311\newstrackr\emailsende.py (3.11.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
25°C
Haze
Search
ENG
IN
09:39
19-11-2022

```
def emailSender(email, token):
    configuration = sib_api_v3_sdk.Configuration()
    configuration.api_key['api-key'] = app.data['mail_api_key']
    api_instance = sib_api_v3_sdk.TransactionalEmailsApi(
        sib_api_v3_sdk.ApiClient(configuration))
    now = datetime.now()
    dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
    msg = {}
    msg['Subject'] = "Verfiy your NewsTracker Account"
    msg['From'] = {"name": "News Tracker Dev Team",
                   "email": "verify@newstracker.com"}
    msg['To'] = [{"email": email}]
    msg['Text']=f'Please click this <a href="http://127.0.0.1:5500/frontend/pages/verify.html?token={token}">link</a> to verify your account'
    html = f"""\
    <html>
        <head></head>
        <body>
        <p>THANK YOU, for joining NewsTracker 🧍</p>
        <br>
        <p>Hurray😃, you just registerd at NewsTracker<br><br>
        Please click the following link to verify your account:<br>
        <a href="http://127.0.0.1:5500/frontend/pages/verify.html?token={token}">Click Here to Verify 😎</a>
        </p>
        <br>
        <p>⚠Note: This link expires within one hour from the time sent</p>
        <br><br>
        <p>Regrads,<br></p>
        <p><a href="https://localhost:5000">NewsTracker Dev Team</a></p>
        <br><br>
        <p>Email sent at {dt_string}</p>
        </body>
    </html>
    """
    send_smtp_email = sib_api_v3_sdk.SendSmtpEmail(
        to=msg['To'], html_content=html, sender=msg['From'], subject=msg['Subject'],text_content=msg['Text'])
    try:
        api_response = api_instance.send_transac_email(send_smtp_email)
        print(api_response)
    except ApiException as e:
        print("Exception when calling SMTPApi->send_transac_email: %s\n" % e)


def newEmailSender(email):
    token = generate_confirmation_token(email)
    emailSender(email, token)
```

7.2 Feature 2

Cookie Checker

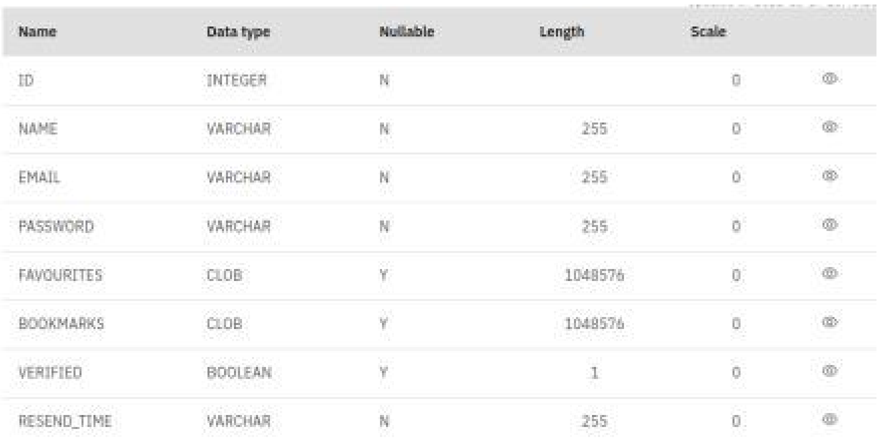

cookiechecker.py - C:\Users\New\AppData\Local\Programs\Python\Python311\newstrackr\cookiechecker.py (3.11.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
25°C
Haze
Search
ENG
IN
09:44
19-11-2022

```
import app
from functools import wraps
import jwt
from flask import request,after_this_request

def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.cookies.get("access_token")
        try:
            data = jwt.decode(token, app.app.config['SECRET_KEY'],algorithms=['HS256'])
            ip=request.headers.get("ip")
            cookieIp=data['ip']
            if(ip!=cookieIp):
                resp={"status":"not logged in"}
                @after_this_request
                def deleter(response):
                    response.delete_cookie("access_token",path="/")
                    response.delete_cookie("email",path="/")
                    return response
                return resp,401
        except:
            resp = {"status":"not logged in"}
            @after_this_request
            def deleter(response):
                response.delete_cookie("access_token",path="/")
                response.delete_cookie("email",path="/")
                return response
            return resp, 401
        return f(data['email'],*args, **kwargs)
    return decorated
```

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

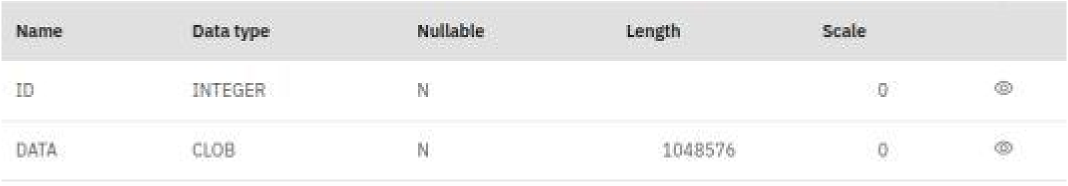
7.3 Database Schema (if Applicable)

User table



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	👁

Bookmarks table



Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	👁
DATA	CLOB	N	1048576	0	👁

8. ADVANTAGES

● This app can be accessed anywhere and anytime, So that the user can view the news

● Its ad free

● The news is based on the API

● The user can bookmark their favourite news.

9. CONCLUSION

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

10. FUTURE SCOPE

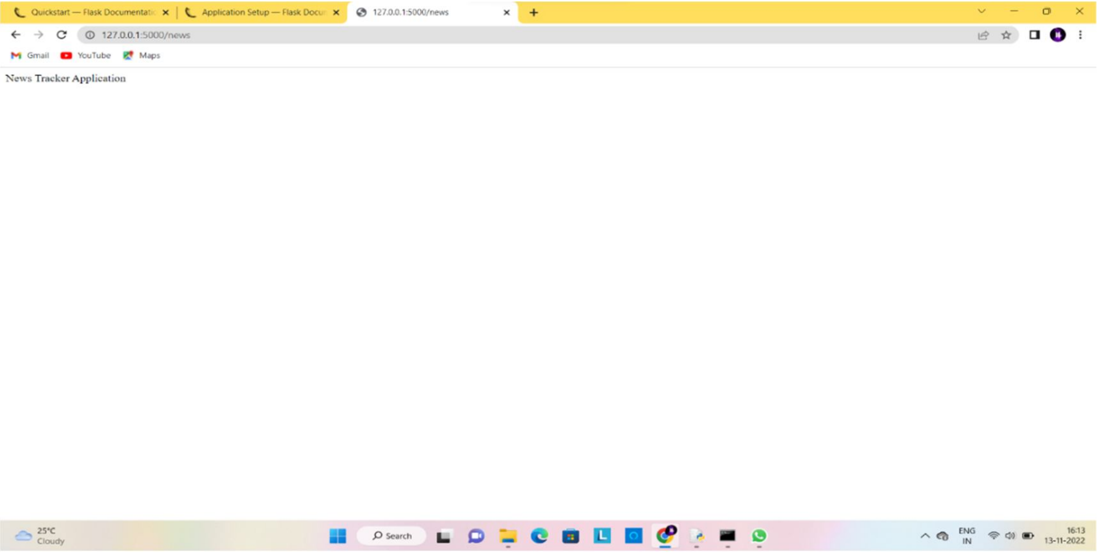
In future we may integrate our own news API instead of third party. APIs and may develop a mobile native application so that it can be used in both android and iOS.

11. APPENDIX

- Source code
  __init__

```
 import os


from flask import Flask




def create_app(test_config=None):

    # create and configure the app

    app = Flask(__name__, instance_relative_config=True)

    app.config.from_mapping(

        SECRET_KEY='dev',

        DATABASE=os.path.join(app.instance_path, 'newstrackr.sqlite'),

    )
```

```
if test_config is None:
    # load the instance config, if it exists, when not testing
    app.config.from_pyfile('config.py', silent=True)
else:
    # load the test config if passed in
    app.config.from_mapping(test_config)

# ensure the instance folder exists
try:
    os.makedirs(app.instance_path)
except OSError:
    pass

# a simple page that says hello
@app.route('/news')
def news():
    return 'News Tracker Application'

return app
```



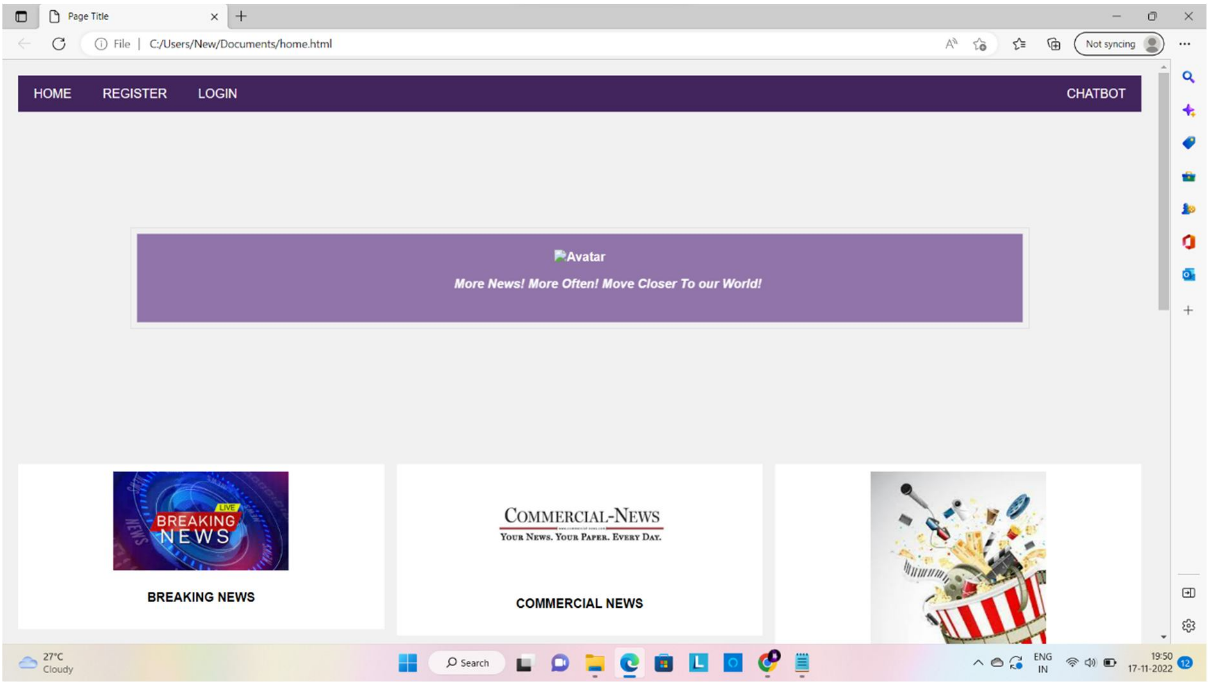
News Tracker Application

Bookmark

```
from flask import request,after_this_request;

from utils.cookieChecker import token_required

from utils.dbQuery import selectQuery,insertQuery


class Bookamark(Resource):

    @token_required

    def get(email,self):

        bookmarks_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE EMAIL=?',(email,))['BOOKMARKS']

        bookmarks_id=bookmarks_id.split(',')

        bookmarks=[]

        if(bookmarks_id==['']):

            bookmarks_id=[]

        for x in bookmarks_id:
            data=selectQuery('SELECT DATA FROM BOOKMARK WHERE
ID=?',(x,))['DATA']
            bookmarks.append(data)
        resp={"data":bookmarks,"id":bookmarks_id}
        return resp,200

    @token_required
    def post(email,self):
        req=request.json
        news=req["news"]
        # Query to check previous inserted
        id=selectQuery('SELECT ID FROM BOOKMARK WHERE DATA=?',(news,))
        if(id==False):
            insertQuery('INSERT INTO BOOKMARK (DATA) VALUES (?)',(news,))
            id=selectQuery('SELECT ID FROM BOOKMARK WHERE
DATA=?',(news,))['ID']
        else:
            id=id['ID']
        new_bookmarks_id=[]
        bookmarks_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE
EMAIL=?',(email,))
        if(bookmarks_id==False):
            bookmarks_id=[]
        else:
            bookmarks_id=bookmarks_id['BOOKMARKS']
            bookmarks_id=bookmarks_id.split(',')
        if(bookmarks_id==['']):
            bookmarks_id=[]
        insertCurr=True
        for x in bookmarks_id:
            new_bookmarks_id.append(x)
            if(int(x)==id):
                insertCurr=False
        if(insertCurr):
            new_bookmarks_id.append(str(id))
            x=",".join([str(i) for i in new_bookmarks_id])
            @after_this_request
            def inserter(response):
                insertQuery('UPDATE USER SET BOOKMARKS=? WHERE
EMAIL=?',(x,email))
                return response
        return {"status":"inserted"},200
class UnBookMark(Resource):
    @token_required
    def post(email,self):
        req=request.json;
        id=req["id"]
        bookmarks_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE
EMAIL=?',(email,))['BOOKMARKS']
        bookmarks_id=bookmarks_id.split(',')
        new_bookmarks=[]
        for x in bookmarks_id:
            if(x!=id):
               new_bookmarks.append(x)
        if(new_bookmarks==[]):
            new_bookmarks=''
        new_bookmarks=",".join([str(i) for i in new_bookmarks])
        @after_this_request
        def inserter(response):
            insertQuery('UPDATE USER SET BOOKMARKS=? WHERE
EMAIL=?',(new_bookmarks,email))
            return response
        return {"status":"deleted"},200
```

Home:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Page Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="style.css">
</head><html>
  <head>
  <meta name="viewport" content="width=device-width, initial-scale=1">

<body>

  <div class="navbar">
    <a href="#">HOME</a>
    <a href="register.html">REGISTER</a>

    <a href="login.html">LOGIN</a>

    <a href="#" class="right">CHATBOT</a>

  </div>


<div class="header">

  <div class="bg-image"></div><div class="bg-text">

    <img src="img.jpeg" alt="Avatar" class="avatar">

    <p><marquee><b><i>More News! More Often!  Move Closer To our World!</i></b></marquee></p>

</div>




  <div class="row">

    <div class="column nature">

      <div class="content">

        <img src="C:\Users\New\Downloads\Breaking news.jpg" alt="breaking news" style="width:50%">

        <h4>BREAKING NEWS</h4>

      </div>

    </div>

    <div class="column nature">

      <div class="content">

      <img src="C:\Users\New\Downloads\Commercial news.png"alt="Lights" style="width:50%">

        <h4>COMMERCIAL NEWS</h4>

      </div>
    </div>
    <div class="column nature">
      <div class="content">
      <img src="C:\Users\New\Downloads\Entertainment news.jpg"alt="Nature"
style="width:50%">
        <h4>ENTERTAINMENT NEWS</h4>
        <p></p>
      </div>
    </div>


    <div class="column cars">
      <div class="content">
        <img src="C:\Users\New\Downloads\Historical news.jpg"alt="Car"
style="width:30%">
        <h4>HISTORICAL NEWS</h4>
      </div>
    </div>
    <div class="column cars">
      <div class="content">
      <img src="C:\Users\New\Downloads\International news.jpg" alt="Car"
style="width:50%">
        <h4>INTERNATIONAL NEWS</h4>
      </div>
    </div>
    <div class="column cars">
      <div class="content">
      <img src="C:\Users\New\Downloads\Local news.jpg" alt="Car"
style="width:50%">
        <h4>LOCAL NEWS</h4>
      </div>
    </div>


    <div class="column people">
      <div class="content">
        <img src="C:\Users\New\Downloads\Sports news.jpg" alt="Car"
style="width:50%">
        <h4>SPORTS NEWS</h4>
      </div>
    </div>
    <div class="column people">
      <div class="content">
      <img src="C:\Users\New\Downloads\weather.jpg" alt="Car"
style="width:50%">
        <h4>WEATHER NEWS</h>
      </div>
    </div>
    <div class="column people">
      <div class="content">
      <img src="C:\Users\New\Downloads\business.jpg"alt="Car"
style="width:100%">
        <h4>BUSINESS NEWS</h4>
      </div>
    </div>
  </div>

  </div>

 <script src="script grid.js"></script>
 <br></div>
<div class="footer">
  <h2>NTAnews</h2>
  <ul>About us</ul>
  <ul>Resouces</ul>
  <ul>Get started</ul>
  <ul>help</ul>
</div>

</body>
</html>
```

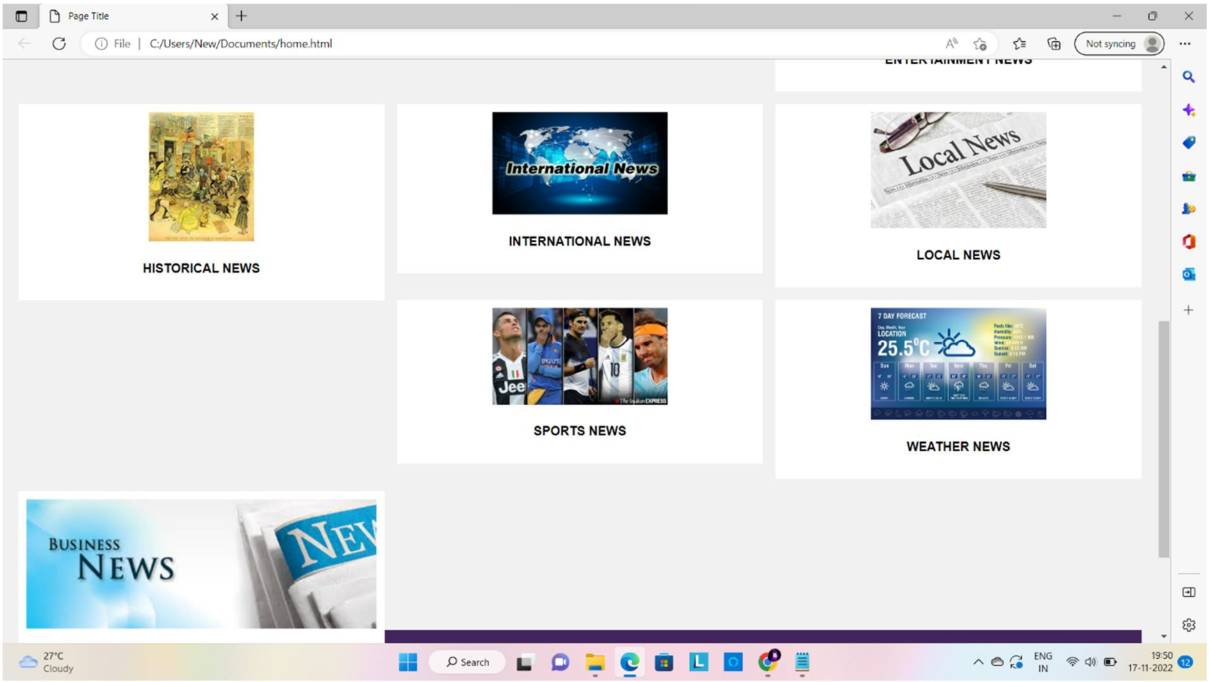


Avatar

More News! More Often! Move Closer To our World!

BREAKING NEWS

COMMERCIAL NEWS



ENTERTAINMENT NEWS

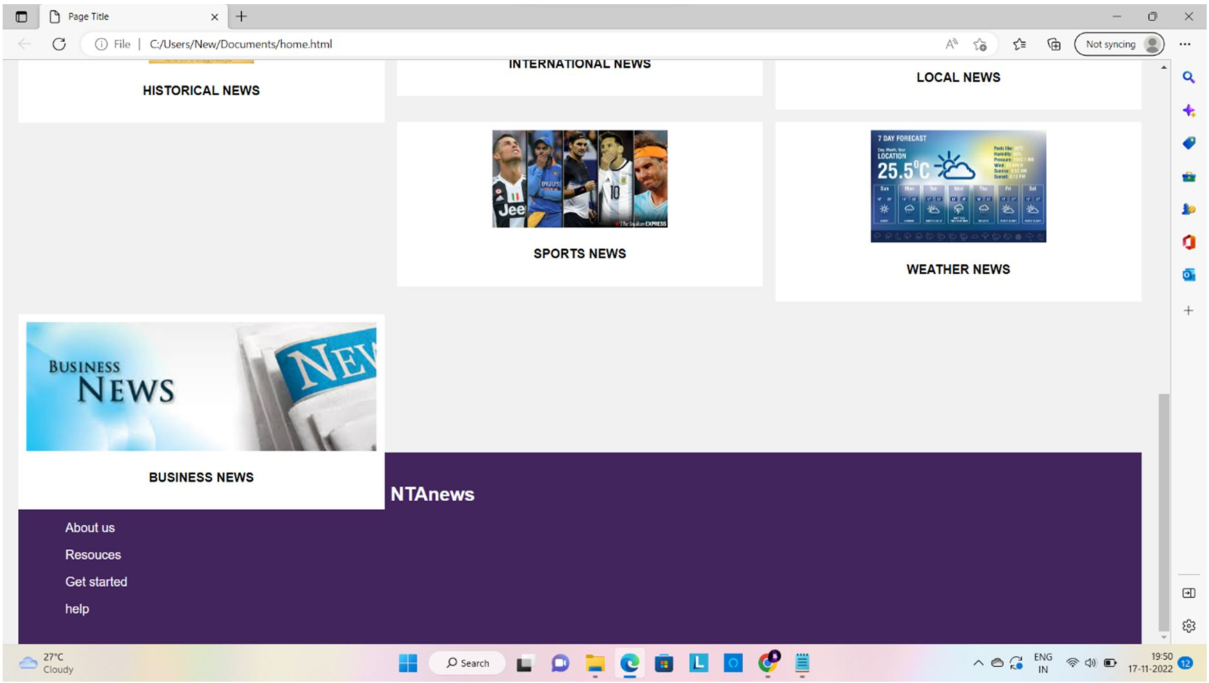
HISTORICAL NEWS

INTERNATIONAL NEWS

LOCAL NEWS

SPORTS NEWS

WEATHER NEWS



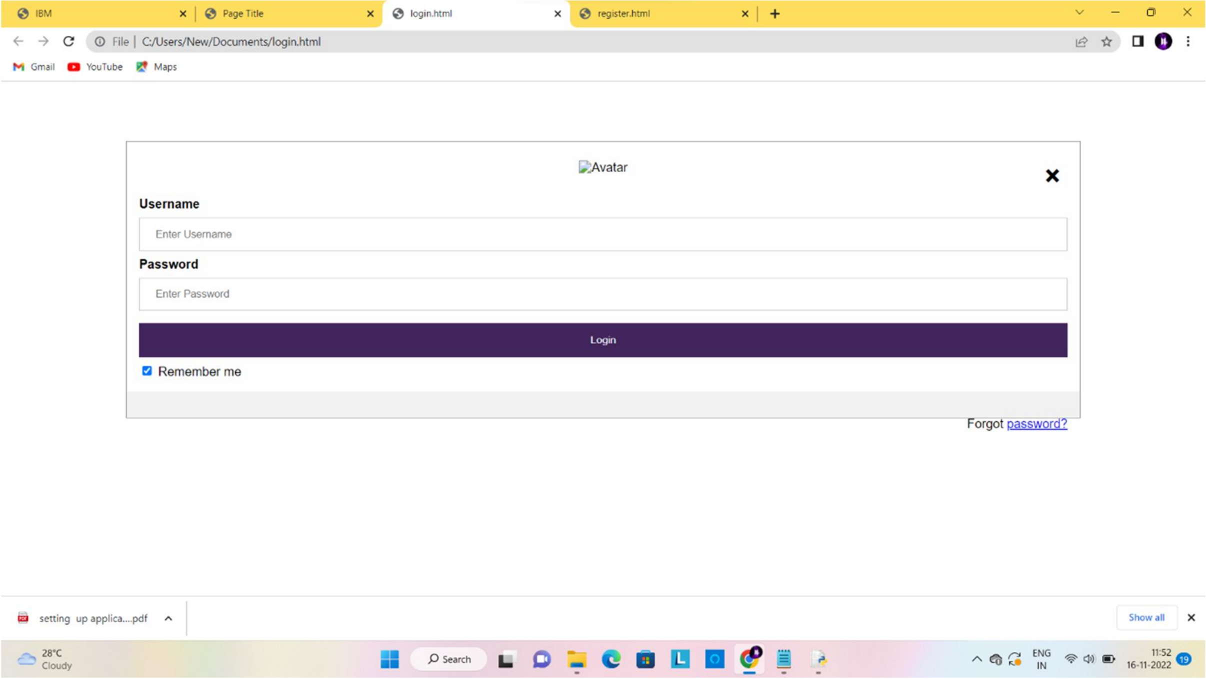
Login:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color:  #43255e;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}
```

```
/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 10%;
  border-radius: 20%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and
centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}
</style>
</head>
<body>

  <form class="modal-content animate" action="/action_page.php"
method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
      <img src="img.jpeg" alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname"
required>

      <label for="psw"><b>Password</b></label>
```

```
   <input type="password" placeholder="Enter Password" name="psw" required>

   <button type="submit">Login</button>
   <label>
    <input type="checkbox" checked="checked" name="remember"> Remember me
   </label>
  </div>

  <div class="container" style="background-color:#f1f1f1">
   <span class="psw">Forgot <a href="#">password?</a></span>
  </div>
 </form>


<script>
var modal = document.getElementById('id01');
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>

</body>
</html>
```



Avatar

Username

Enter Username

Password

Enter Password

Login

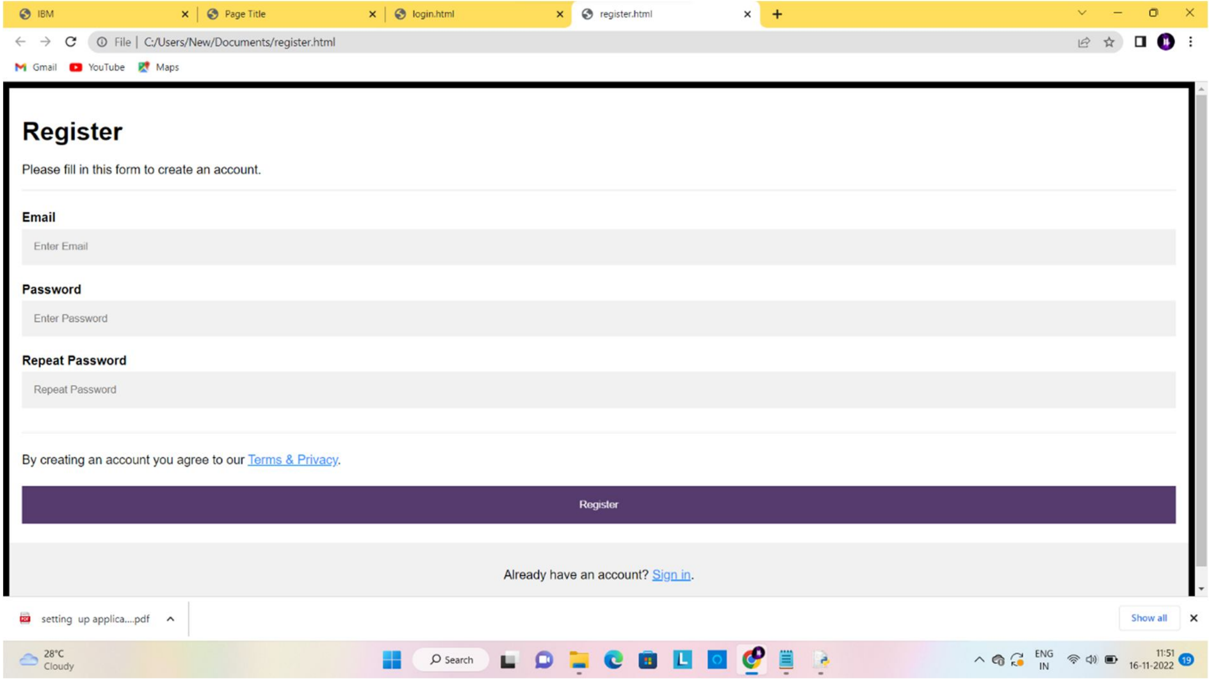
☑ Remember me

Forgot password?

Register:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-color: black;
}

* {
  box-sizing: border-box;
```

```
}

/* Add padding to containers */
.container {
  padding: 16px;
  background-color: white;
}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit button */
.registerbtn {
  background-color: #43255e;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

.registerbtn:hover {
  opacity: 1;
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
}
</style>
</head>
<body>

<form action="/action_page.php">
  <div class="container">
    <h1>Register</h1>
    <p>Please fill in this form to create an account.</p>
    <hr>

    <label for="email"><b>Email</b></label>
    <input type="text" placeholder="Enter Email" name="email" id="email" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" id="psw" required>

    <label for="psw-repeat"><b>Repeat Password</b></label>
    <input type="password" placeholder="Repeat Password" name="psw-repeat" id="psw-repeat" required>
    <hr>
    <p>By creating an account you agree to our <a href="#">Terms & Privacy</a>.</p>
    <button type="submit" class="registerbtn">Register</button>
  </div>

  <div class="container signin">
    <p>Already have an account? <a href="login.html">Sign in</a>.</p>
  </div>
</form>

</body>
</html>
```



Register

Please fill in this form to create an account.

Email

Enter Email

Password

Enter Password

Repeat Password

Repeat Password

By creating an account you agree to our Terms & Privacy.

Register

Already have an account? Sign in.