

IBM Experiential Learning
NEWS TRACKER APPLICATION

Team ID:2022TMID04881

Surya Prasaath L 19ITR098

Surya Narayanan D 19ITR097

Swetha R 19ITR099

Tamilmani K 19ITR100

Index

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

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 Link

1. Introduction

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.

a. Project Overview

- The user interacts with the application.
- Registers by giving the details.
- Integrate the application with news APIs and store the data in the database.
- The database will have all the details and the user can search the news by using a search bar.

b. Purpose

A news application is a big interactive database that tells a news story. Think of it like you would any other piece of journalism. It just uses software instead of words and pictures.

2. Literature Survey

Web tracking seems to become ubiquitous in online business and leads to increased privacy concerns of users. This paper provides an overview over the current state of the art of web-tracking research, aiming to reveal the relevance and methodologies of this research area and create a foundation for future work. In particular, this study addresses the following research questions: What methods are followed? What results have been achieved so far? What are potential future research areas? For these goals, a structured literature review based upon an established methodological framework is conducted. The identified articles are investigated with respect to the applied research methodologies and the aspects of web tracking they emphasize.

2.1 Existing problem

There are lot of time consumption to read the relevant news that we need without categorising. With the help of categorising we can able to read the news which we need . 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.

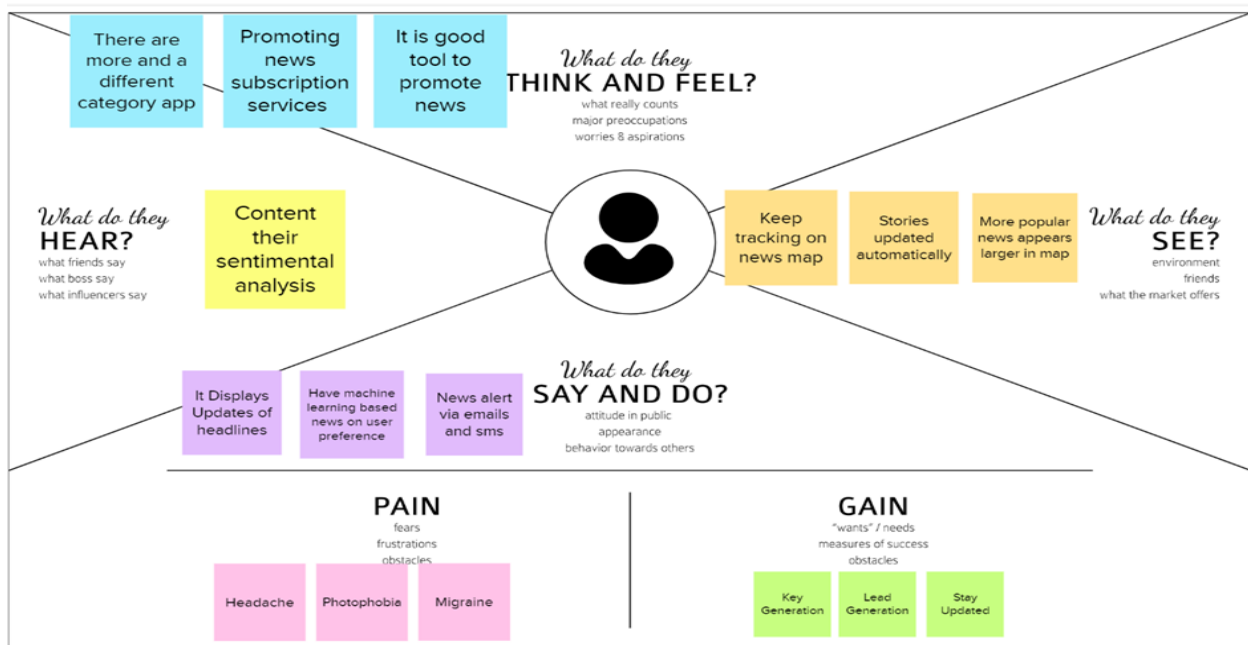
2.2 References

2.3 Problem Statement Definition

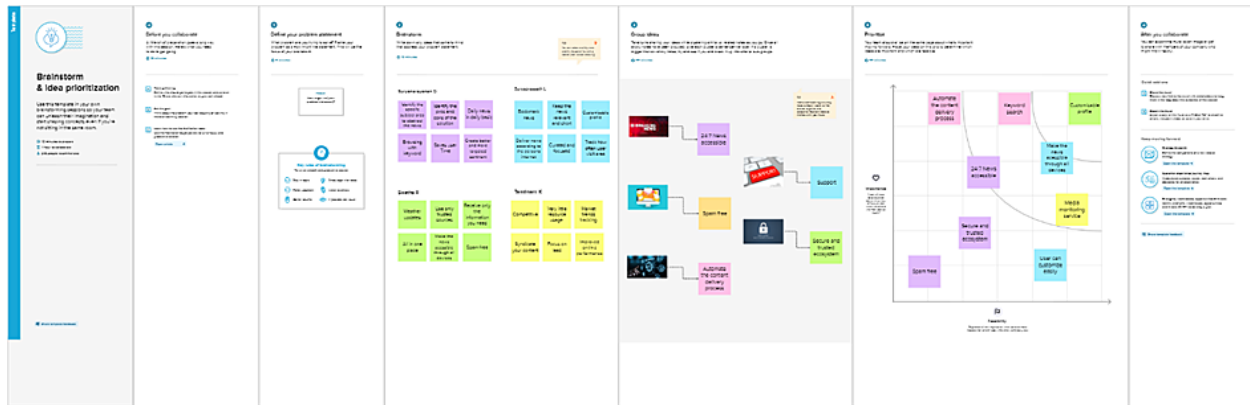
There are multiple news sharing apps used by single user are often spammed with notifications. There is also a lot of news with different categories which gets shared .A news sharing app wants to help users find relevant and important news easily.

3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canva



3.2 Ideation & Brainstorming



3.3 Proposed Solution

The main focus of this application is to connect news article from all around the world and deliver it to user as fast as possible in best visualise way.

3.4 Problem Solution fit

Project Design Phase-I - Solution Fit

Project Title: NEWS TRACKER APPLICATION

Team ID: PNT2022TMID04881

<p>Define CS, fit into CC</p> <p>1. CUSTOMER SEGMENT(S) CS</p> <ul style="list-style-type: none"> Marketing people General people Digital makeover lover 	<p>6. CUSTOMER CONSTRAINTS CC</p> <ul style="list-style-type: none"> Account security ? Ability to adapt to different orientation ? 	<p>5. AVAILABLE SOLUTIONS AS</p> <p>When a user is not online due to some reason should have to access to the internet. Whenever the user is online the news content is downloaded in the cache memory of the app, this is how a user can access to the content offline. The option of filtering the content based on different category</p> <p>Explore AS, differentiate</p>
<p>Focus on JAP, fit into BE, understand RC</p> <p>2. JOBS-TO-BE-DONE / PROBLEMS</p> <ul style="list-style-type: none"> decide what, where and when to change for better performance want to keep financial details private and prevent leakage 	<p>9. PROBLEM ROOT CAUSE RC</p> <p>At this digital age, people are captivated by smartphones or other mobile devices. They want all necessitous info to be found at their fingertips.</p>	<p>7. BEHAVIOUR BE</p> <p>People need real-time news and updates immediately so the news app has a very bright future. A news app that provides a dynamic experience will get a huge success.</p> <p>Focus on JAP, fit into BE, understand RC</p>

3. TRIGGERS News app triggers to describe automatically specific news or event faster and more targeted	10. YOUR SOLUTION Evident that developing a news app helps you stand out among all competitors. The aforesaid aspects let a news app promote an online magazine and as a result, the app reaches out and attracts many people from all across the world.	8.CHANNELS of BEHAVIOUR 8.1 ONLINE User can access all kind of news in many forms 8.2 OFFLINE User can view news in offline mode
---	---	---

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

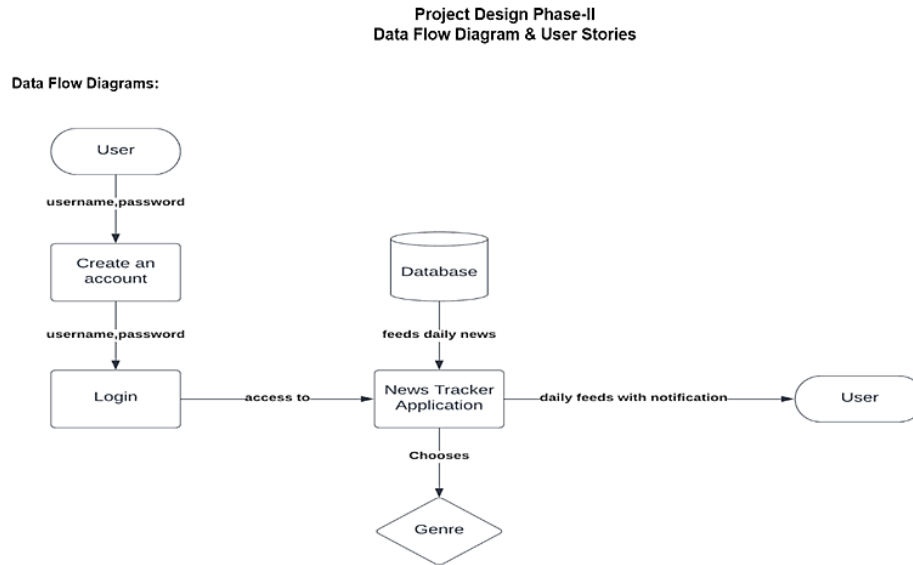
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
Fr-1	UserRegistration	Registration through Form Registration through Gmail Registration through LinkedIN
Fr-2	User Confirmation	Confirmation via Email Confirmation via OTP
Fr-3	News Publication	Login Sign Up
Fr-4	Favorite news genre suggestion	Follow or Subscribe the related topics

4.2 Non-Functional requirements

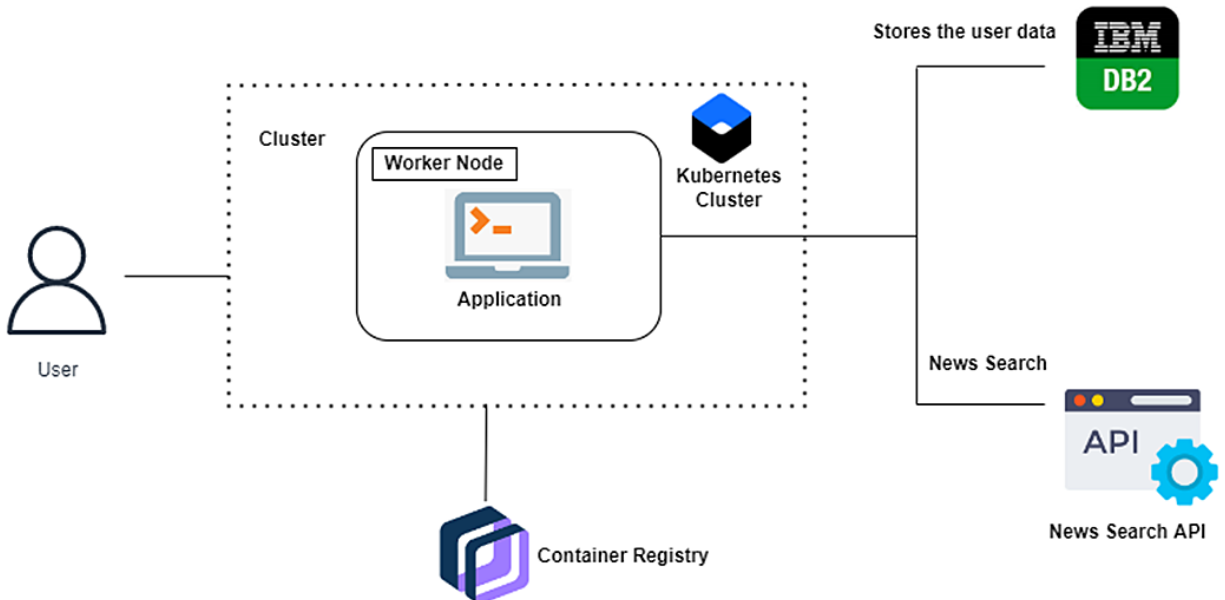
FR No.	Non-Functional Requirement	Description
NFr-1	Usability	Daily Usage for regular use
NFr-2	Security	Based on the userregistration
NFr-3	Reliability	100% reliable our privacy policyis for usersecurity
NFr-4	Performance	Performance will be as good as any othernews application app
NFr-5	Availability	It is open sourceand free appor free account tocreate and read news
NFr-6	Scalability	Provides excellent scalability in terms of valuablenews

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirements	User Story Number	User Story / Task	Acceptance Criteria	Priority
Customer (Mobile User)	Registration	USN-1	As a user, I can register to the application using my email and password	I can access my account / application	High
	Login	USN-2	As a user, I can login to the application using my email and password		Medium
	Dashboard / Application				
Administrator					
Customer Care Executive					

6. PROJECT PLANNING & SCHEDULING

6.1 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.	10	High	SURYA PRASAATH
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	SURYA NARAYANAN
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password.	15	High	SWETHA
Sprint-2	Input Necessary Details	USN-4	As a user, I can search the news in the application	15	High	TAMIL MANI
Sprint-2	Data Pre-processing	USN-5	The application searches for news related to the entered details.	15	High	TAMIL MANI
Sprint-3	Searching of news	USN-6	As a user, I can search for the accurate news what I want	20	High	SURYA PRASAATH
Sprint-3		USN-7	As a user, I can get accurate news in the application	5	Medium	SURYA NARAYANAN
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	SWETHA

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	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

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

```
from flask import Flask,render_template,request,url_for,redirect,flash,session
import hashlib
import ibm_db
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import requests

app=Flask(__name__)
app.secret_key = "abc"

#cloud
connection = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=bky07012;PWD=OKookhO0dUuBjV
9t;",",")

news_api_key="f71565537970409cbe006afcbac30c33"

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

@app.route("/signup")
def signup():
```

```

return render_template("signup.html")

@app.route("/register",methods=['GET','POST'])
def register():
    error = None

    if request.method=='POST':
        username= request.form['username']
        email=request.form['email']
        password=request.form['password']
        cpass=request.form['cpassword']

        if password != cpass:
            error = "Password and Confirm password should be same!!"
        else:
            hashed_password=hashlib.sha256(password.encode()).hexdigest()

            sql="SELECT * FROM newstracker WHERE username=?"
            stat =ibm_db.prepare(connection,sql)
            ibm_db.bind_param(stat,1,username)
            ibm_db.execute(stat)
            res=ibm_db.fetch_assoc(stat)

            if res:
                error="Username is already exists user different username"

            else:

                sql1="INSERT INTO newstracker VALUES (?,?)"
                pre_stat =ibm_db.prepare(connection,sql1)
                ibm_db.bind_param(pre_stat,1,username)

```

```
        ibm_db.bind_param(pre_stat,2,email)
        ibm_db.bind_param(pre_stat,3,hashed_password)
        ibm_db.execute(pre_stat)

        flash('Account created successfully')
        return redirect(url_for('index'))
    return render_template('signup.html', error = error)
```

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

```
def login():
```

```
    error = None
```

```
    if request.method=="POST":
```

```
        username=request.form['username']
```

```
        password=request.form['password']
```

```
        hashed=hashlib.sha256(password.encode()).hexdigest()
```

```
    sql="SELECT * FROM newstracker WHERE username=?"
```

```
    con =ibm_db.prepare(connection,sql)
```

```
    ibm_db.bind_param(con,1,username)
```

```
    ibm_db.execute(con)
```

```
    res=ibm_db.fetch_assoc(con)
```

```
    if res:
```

```
        if hashed == res['PASSWORD']:
```

```
            session['username'] = request.form['username']
```

```
        return redirect(url_for('home'))

    else:

        error = "Login Failed!!"

        return render_template('index.html',error=error)

    else:

        error = "Login Failed!!"

        return render_template('index.html',error=error)
```

```
@app.route('/home')

def home():

    error=None

    if 'username' in session:

        username = session['username']

        main_url="https://newsapi.org/v2/top-headlines?country=in&apiKey="+news_api_key

        news=requests.get(main_url).json()

        articles=news["articles"]

        news_articles_title=[]

        for a in articles:

            news_articles_title.append(a["title"])


        news_articles_description=[]

        for a in articles:

            news_articles_description.append(a["description"])
```

```
news_articles_url=[]
```

```
for a in articles:
```

```
    news_articles_url.append(a["url"])
```

```
news_articles_urlToImage=[]
```

```
for a in articles:
```

```
    news_articles_urlToImage.append(a["urlToImage"])
```

```
news_articles_publishedAt=[]
```

```
for a in articles:
```

```
    news_articles_publishedAt.append(a["publishedAt"])
```

```
news_articles_content=[]
```

```
for a in articles:
```

```
    news_articles_content.append(a["content"])
```

```
    return render_template("home.html",user=username,articles=articles)
```

```
else:
```

```
    error="Pleasr Login to access"
```

```
    return render_template('index.html',error=error)
```

```
@app.route('/sports')
```

```
def sports():
```

```
    error=None
```

```
    if 'username' in session:
```

```
        username = session['username']
```

```
    main_url="https://newsapi.org/v2/top-
headlines?country=in&category=sports&apiKey="+news_api_key

    news=requests.get(main_url).json()
    articles=news["articles"]

    return render_template("home.html",user=username,articles=articles)
else:
    error="Pleasr Login to access"

    return render_template('index.html',error=error)
```

```
@app.route('/entertainment')
```

```
def entertainment():
```

```
    error=None
```

```
    if 'username' in session:
```

```
        username = session['username']
```

```
        main_url="https://newsapi.org/v2/top-
headlines?country=in&category=entertainment&apiKey="+news_api_key
```

```
        news=requests.get(main_url).json()
```

```
        articles=news["articles"]
```

```
        return render_template("home.html",user=username,articles=articles)
```

```
    else:
```

```
        error="Pleasr Login to access"
```

```
        return render_template('index.html',error=error)
```

```
@app.route('/technology')
```

```
def technology():
```

```
    error=None
```



```

if 'username' in session:

    username = session['username']

    main_url="https://newsapi.org/v2/top-
headlines?country=in&category=technology&apiKey="+news_api_key

    news=requests.get(main_url).json()

    articles=news["articles"]

    return render_template("home.html",user=username,articles=articles)

else:

    error="Pleasr Login to access"

    return render_template('index.html',error=error)

```

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

```
def science():
```

```
    error=None
```

```
    if 'username' in session:
```

```
        username = session['username']
```

```
        main_url="https://newsapi.org/v2/top-
headlines?country=in&category=science&apiKey="+news_api_key
```

```
        news=requests.get(main_url).json()
```

```
        articles=news["articles"]
```

```
        return render_template("home.html",user=username,articles=articles)
```

```
    else:
```

```
        error="Pleasr Login to access"
```

```
        return render_template('index.html',error=error)
```

```
@app.route('/business')
```

```
def business():
```

```

error=None

if 'username' in session:

    username = session['username']

    main_url="https://newsapi.org/v2/top-
headlines?country=in&category=business&apiKey="+news_api_key

    news=requests.get(main_url).json()

    articles=news["articles"]

    return render_template("home.html",user=username,articles=articles)

else:

    error="Please Login to access"

    return render_template('index.html',error=error)

```

```

@app.route('/world')

```

```

def world():

    error=None

    if 'username' in session:

        username = session['username']

        main_url="https://newsapi.org/v2/top-headlines?sources=bbc-
news&apiKey="+news_api_key

        news=requests.get(main_url).json()

        articles=news["articles"]

        return render_template("home.html",user=username,articles=articles)

    else:

        error="Pleasr Login to access"

        return render_template('index.html',error=error)

```

```

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

```

```
def search():
    if 'username' in session:
        username = session['username']
        if request.method=="POST":
            search=request.form["search"]
            news_data =
requests.get(f'https://newsapi.org/v2/everything?q={search}&apiKey='+news_api_key).json()
            articles=news_data['articles']
            return render_template("home.html",user=username,articles=articles)
```

```
@app.route('/logout')
```

```
def logout():
    session.pop('username', None)
    return redirect(url_for('index'))
```

```
@app.route('/forgot_password')
```

```
def forgot_password():
    return render_template("resetpassword.html")
```

```
@app.route("/verify_email",methods=["GET","POST"])
```

```
def verify_email():
    error=None
    if request.method=="POST":
        email=request.form["email"]
```

```
sql="SELECT * FROM newstracker WHERE email=?"
```

```
stat =ibm_db.prepare(connection,sql)
```

```
ibm_db.bind_param(stat,1,email)
```

```
ibm_db.execute(stat)
```

```
res=ibm_db.fetch_assoc(stat)
```

```
if res:
```

```
    return render_template("resetpassword.html",result="200",email=email)
```

```
else:
```

```
    error="No such email is available"
```

```
    return render_template("resetpassword.html",result="404",error=error)
```

```
@app.route("/update_password",methods=["GET",'POST'])
```

```
def update_password():
```

```
    error=None
```

```
    if request.method=="POST":
```

```
        password=request.form["newp"]
```

```
        cpass=request.form["cp"]
```

```
        email=request.form["email"]
```

```
        if cpass==password:
```

```
            hashed=hashlib.sha256(password.encode()).hexdigest()
```

```
            sql="UPDATE newstracker SET password=? WHERE email=?"
```

```
            stat =ibm_db.prepare(connection,sql)
```

```
            ibm_db.bind_param(stat,1,hashed)
```

```
            ibm_db.bind_param(stat,2,email)
```

```
            ibm_db.execute(stat)
```

```

        error="Password reset successful"

        return render_template("index.html",error=error)
    else:

        error="Password and confirm password must be same"

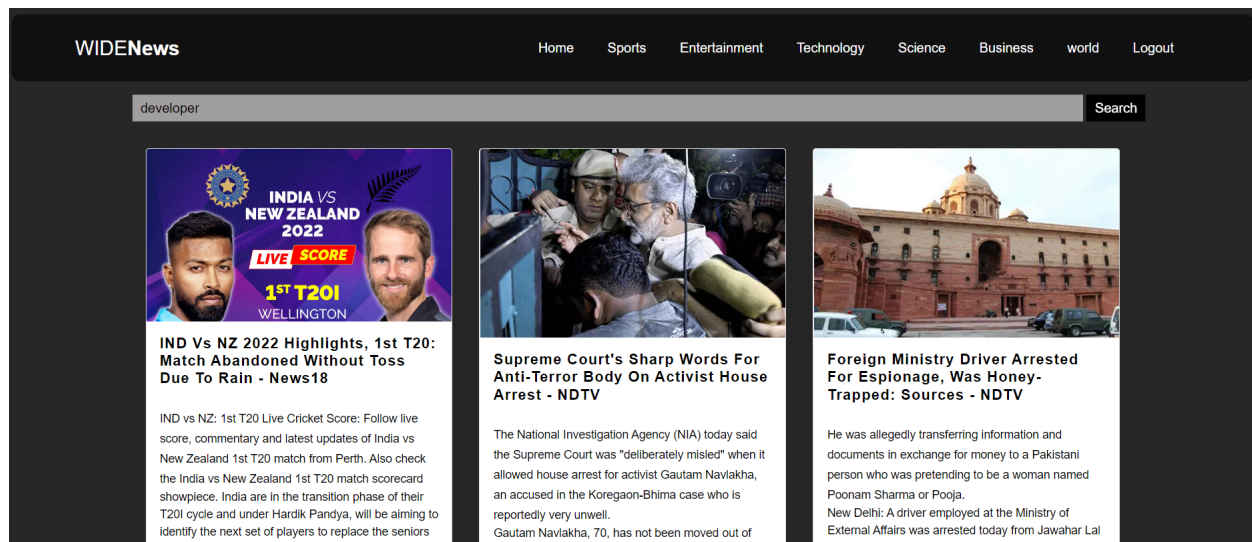
        return render_template("resetpassword.html",error=error)

if __name__=="__main__":

    app.run(debug=True)

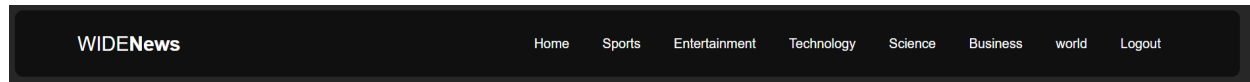
```

7.1 Feature 1



- **User can search required news**
- **Find particular news**

7.2 Feature 2



➤ **Category wise news are available**

7.3 Database Schema (if Applicable)

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

BKY07012.NEWSTRACKER

Back

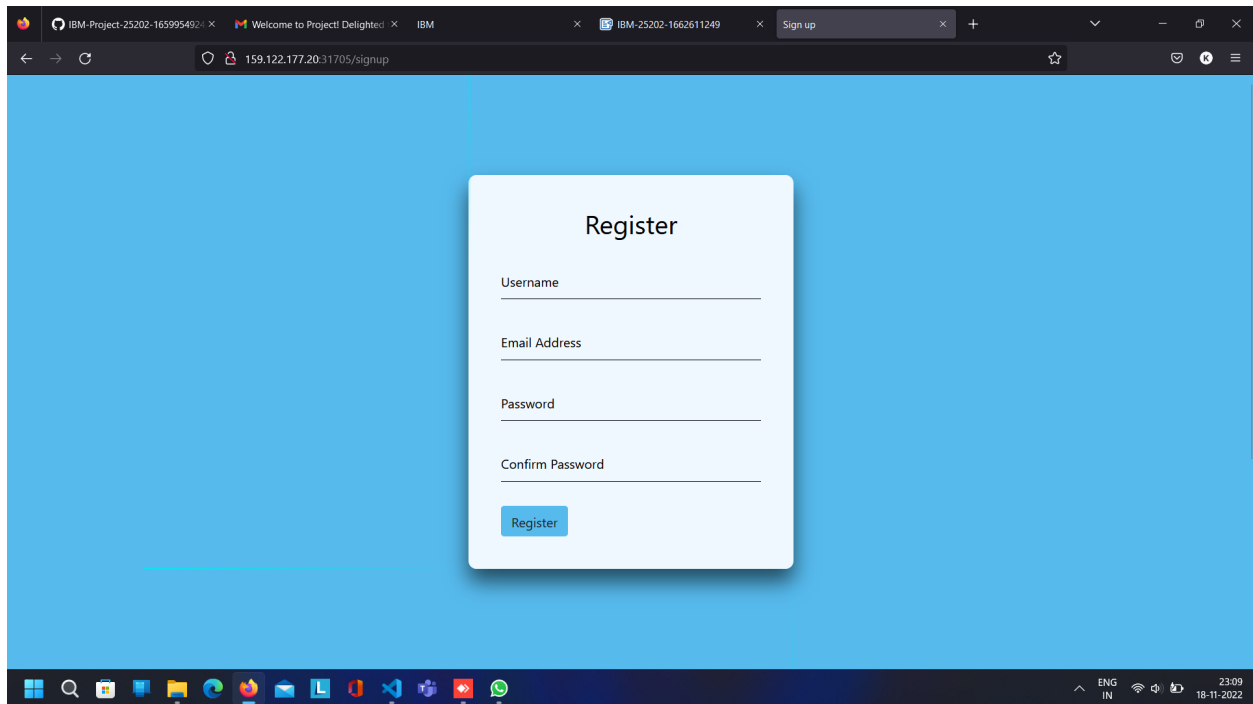
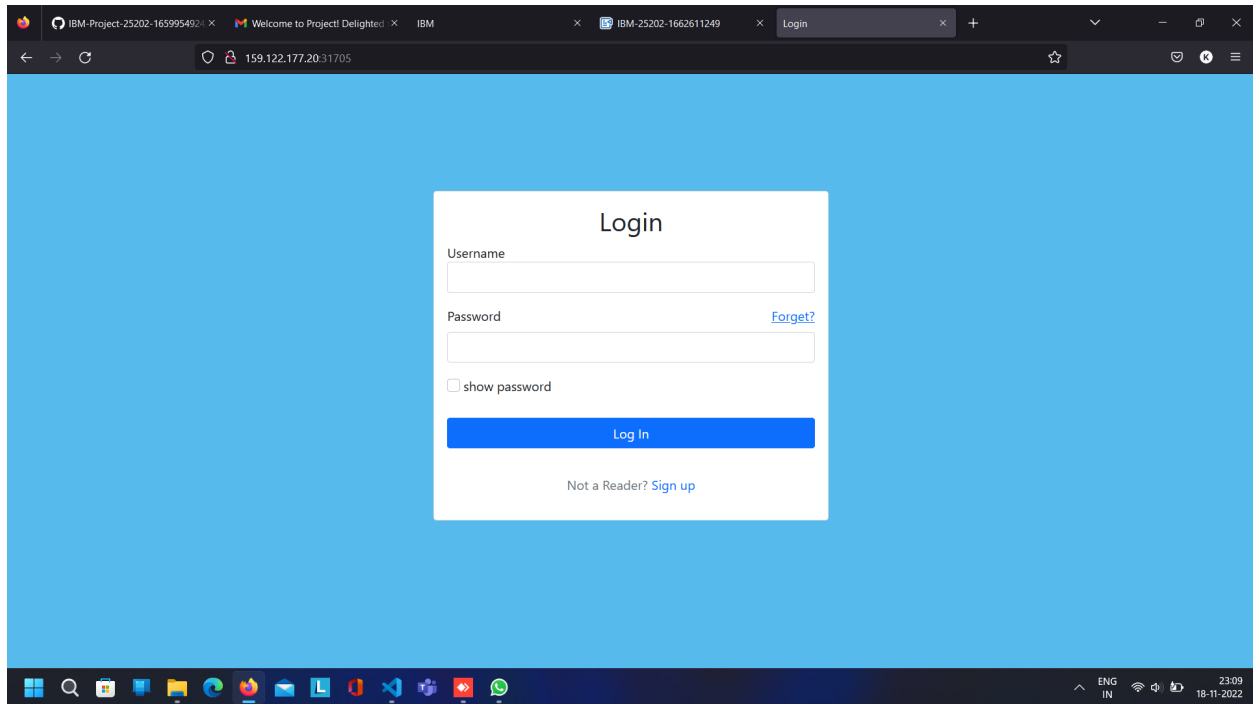
Export to CSV

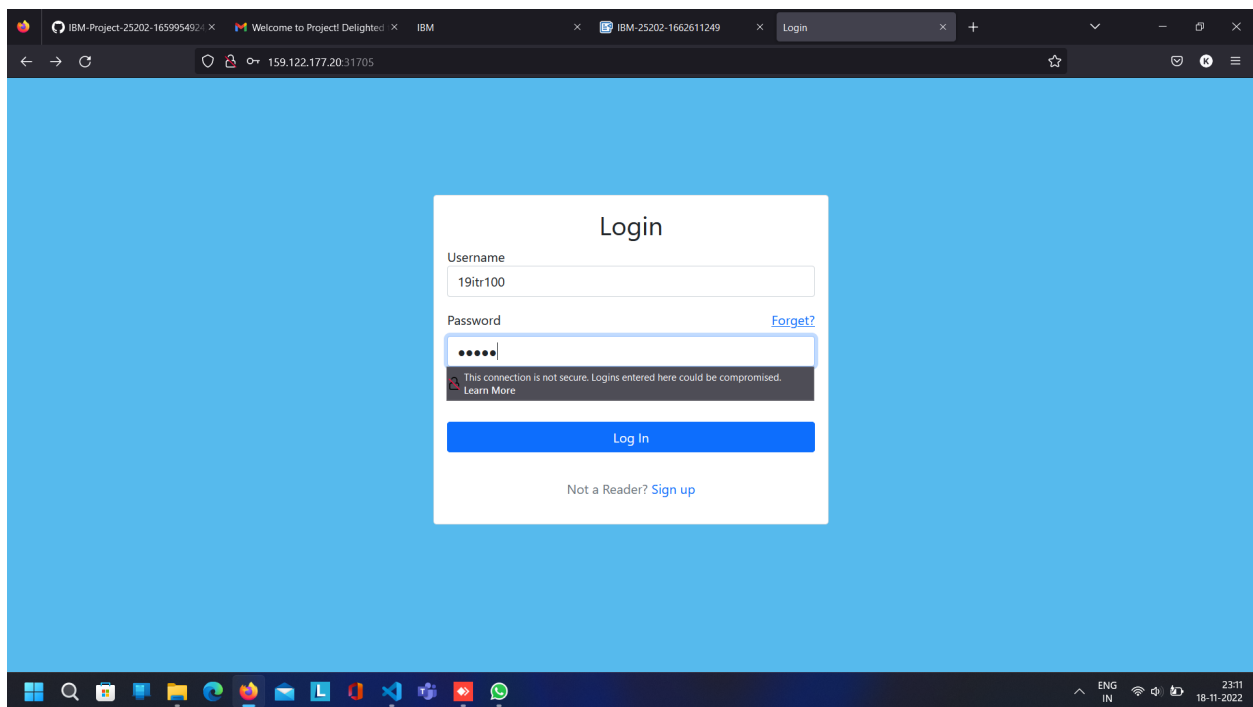
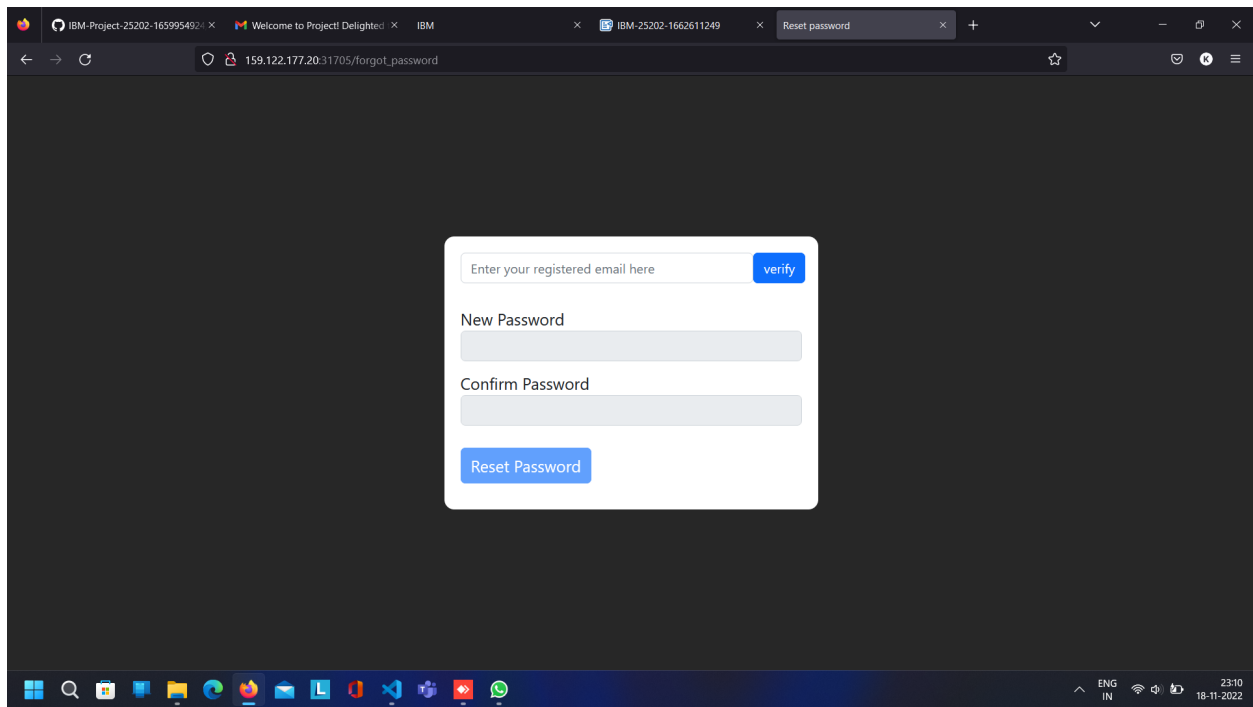
USERNAME	EMAIL	PASSWORD
tamilmani	k.tamilmani1234@gmail.com	d9614030c34f73f1b9b755459977284960c50b37605262fe9828cad6158cf8a7

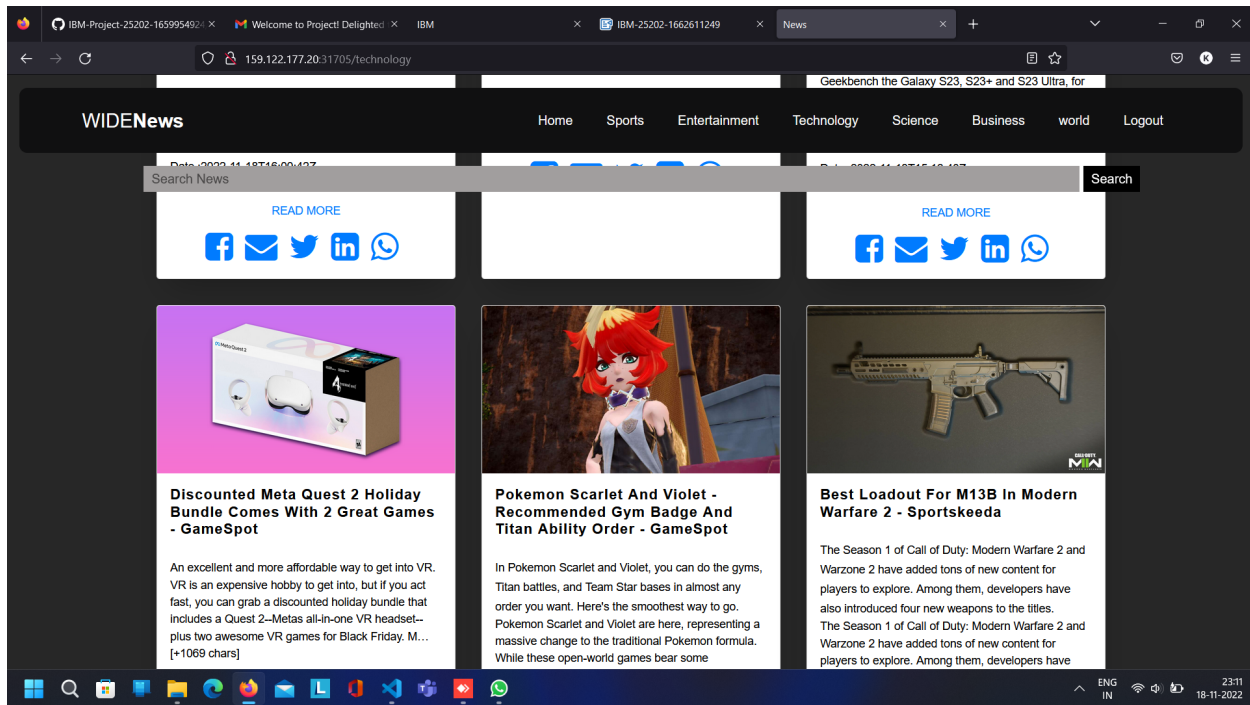
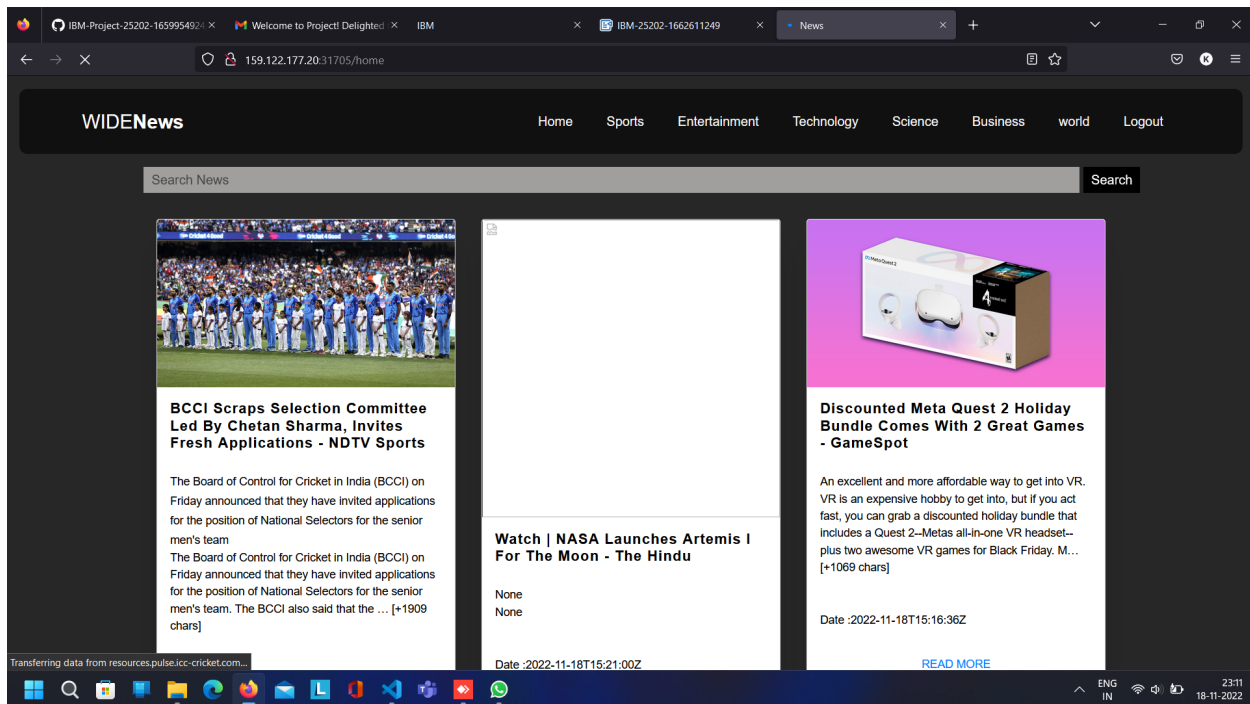
30°C Mostly sunny 20:49 18-11-2022

8. RESULTS

8.1 Performance Metrics







9.1 ADVANTAGES

This app saves lots of time by categorising the news which makes the user to read the relevant news they want.

9.2 DISADVANTAGES

The user always requires Internet Connection to use this Application.

By categorising the news, the user could not be able to access other important news that happens.

10. CONCLUSION

Hence this application satisfies the user and encourages them to study the news. They can be able to read the news from compact mobile phones so they no need to carry the newspapers to every place they go. With good interface the users are satisfied. By collecting the requirements, the users are provided the news categories that they can choose to need. And also the search option is provided for user to search for the particular news they need.

This application is reliable with good user interface. The application is secure with the Secure Sockets Layer. The use of IBM cloud makes the application available in all places including remote areas with the presence of good internet facility.

11. FUTURE SCOPE

Providing the facility to the user to choose the country the need to know. Locating the location of the user and providing the news based on his/her location. And changing of news data if the user moves from one location to other. And providing the facility to the user to read the news offline. The security of the user data can be improved. The user can be verified before registering.

13. APPENDIX

Source Code: <https://github.com/IBM-EPBL/IBM-Project-25202-1659954924>

GitHub & Project Demo :

https://drive.google.com/file/d/1ekMaNJvsU2anFGm4ju68jHiZgow6433U/view?usp=share_link