



SARANATHAN COLLEGE OF ENGINEERING
VENKATESHWARA NAGAR, PANJAPPUR, TRICHY-12

NALAIYA THIRAN

PROJECT TITLE : NEWS TRACKER APPLICATION

DOMAIN : CLOUD COMPUTING

TEAM ID : PNT2022TMID32880

TEAM LEADER :

SURIYANAD S - 813819205058@smartinternz.com

TEAM MEMBERS :

GANESH V - 813819205017@smartinternz.com

MOHAN RAM M - 813819205037@smartinternz.com

SEENI ARIVAZHAKAN - 813819205053@smartinternz.com

GITHUB : <https://github.com/IBM-EPBL/IBM-Project-16312-1659611345>

BONAFIDE CERTIFICATE

Certified that this project report “**NEWS TRACKER APPLICATION**”
is the bonafide work of

SURIYANAD S	-	813819205058
GANESH V	-	813819205017
MOHAN RAM M	-	813819205037
SEENI ARIVAZHAKAN	-	813819205053

who carried out the project under my supervision

SIGNATURE

Dr. R Thillaikarasi.,

HEAD OF THE DEPARTMENT

Professor,

Information Technology,

Saranathan College of Engineering,

Tiruchirappalli - 620 012

SIGNATURE

Ms.N.Bhavani, M.E.,

EVALUATOR

Associate Professor,

Information Technology,

Saranathan College of Engineering,

Tiruchirappalli - 620 012

SIGNATURE

Ms.J.Sangeethapriya, M.Tech.,

MENTOR

Assistant Professor,

Information Technology,

Saranathan College of Engineering,

Tiruchirappalli - 620 012

TABLE OF CONTENTS

S. No	CONTENT	PAGE NO
	LIST OF TABLES	V
	LIST OF FIGURES	VI
	ABSTRACT	VII
1	INTRODUCTION	1
	1.1 PROJECT OVERVIEW	1
	1.2 PURPOSE	1
2	LITERATURE SURVEY	2
	2.1 EXISTING PROBLEMS	2
	2.2 REFERENCES	3
	2.3 PROBLEM STATEMENT DEFINITION	3
3	IDEATION & PROPOSED SOLUTION	4
	3.1 EMPATHY MAP CANVAS	4
	3.2 IDEATION & BRAINSTORMING	5
	3.3 PROPOSED SOLUTION	6
	3.4 PROBLEM SOLUTION FIT	7
4	REQUIREMENTS ANALYSIS	8
	4.1 FUNCTIONAL REQUIREMENT	8
	4.2 NON-FUNCTIONAL REQUIREMENTS	8
5	PROJECT DESIGN	9
	5.1 DATA FLOW DIAGRAM	9
	5.2 SOLUTION & TECHNICAL ARCHITECTURE	9
	5.3 USER STORIES	10
6	PROJECT PLANNING & SCHEDULING	11
	6.1 SPRINT PLANNING & ESTIMATION	11
	6.2 SPRINT DELIVERY SCHEDULE	11
	6.3 REPORTS FROM JIRA	12
7	TESTING	13
	7.1 TEST CASES	13

8	OUTPUT	14
	8.1 LOGIN PAGE	14
	8.2 REGISTER PAGE	14
	8.3 HOME PAGE	15
9	ADVANTAGES AND DISADVANTAGES	17
10	CONCLUSION	18
11	FUTURE SCOPE	18
12	APPENDIX	18
	12.1 SOURCE CODE	18
	12.2 PROJECT GITHUB LINK	26
	12.3 PROJECT DEMO LINK	26

LIST OF TABLES

TABLE. No	NAME OF THE TABLE	PAGE No
TABLE 3.1	PROPOSED SOLUTIONS	6
TABLE 4.1	FUNCTIONAL REQUIRMENT	8
TABLE 4.2	NON-FUNCTIONAL REQUIRMENTS	8
TABLE 5.1	USER STORIES	10
TABLE 6.1	SPRINT PLANING & ESTIMATION	11
TABLE 6.2	SPRINT DELIVERY SHEDULE	11
TABLE 7.1	TEST CASES	13

LIST OF FIGURES

FIGURE No	NAME OF THE FIGURE	PAGE No.
FIGURE 3.1	EMPATHY MAP	4
FIGURE 3.2	BRAINSTROMING	5
FIGURE 3.3	PROBLEM SOLUTION SET	7
FIGURE 5.1	DATA FLOW DIAGRAM	9
FIGURE 5.2	TECHNICAL ARCHITECTURE	10
FIGURE 6.1	BURNDOWN CHART	12
FIGURE 8.1	LOGIN PAGE	14
FIGURE 8.2	REGISTER PAGE	14
FIGURE 8.3	TODAY'S NEWS	15
FIGURE 8.4	BUISNESS NEWS	15
FIGURE 8.5	DATABASE CONNECTION	16

ABSTRACT

News is defined as “the trustworthy piece of information on a recent event that occurred “. Around the world there are plenty of occurrences are happening and a lot of news is created, telecasted and printed. But according to a survey on average, 45.43% people believe they encounter fake news online daily. So, because of this around a day, the person reading/watching the news is gradually decreasing.

Even though if they would watch/read the news there is much false news and clickbait notifications make them awful. Even on television channels, we don’t know whether the news they are showing is true or not. This affects a normal person’s daily updates. So we have brought news from the trusted sources and displayed it using our “News App”.

This application majorly helps the user by offering piece of crisp news so that the user could have enough time to spend reading news. It also offers different sections of news the user could select and read their favorite section news.

1. INTRODUCTION

News is information about current events. This may be provided through many different media: word of mouth, printing, postal systems, broadcasting, electronic communication, or through the testimony of observers and witnesses to events. Throughout history, people have transported new information through oral means. Having developed in China over centuries, newspapers became established in Europe during the early modern period. In the 20th century, radio and television became an important means of transmitting news. Whilst in the 21st, the internet has also begun to play a similar role.

But nowadays people have no time to read news, so we have decided to create a News application which makes users to read news in ease.

1.1 Project Overview

Over the period of years, people have updated their way of news consumption. Right from the traditional way of morning newspapers till easy updates on their mobile phones, technology has given a new way to the news publishing industry. If you are running a news publisher and wish to expand in the digital world, you have landed at the right place.

As more and more people are shifting from print media to digital news sources, the competition between the online streaming platforms is also increasing. To stand out from the competition, it is essential to develop a news app that provide the listed major factors:

Comfort: Ease while browsing the news updates of their choice.

Convenience: Accessing the news in poor or bad internet connectivity.

Control: Getting updates from various desired categories rather than unwanted updates.

1.2 Purpose

Know fundamental concepts and can work on IBM Cloud and Db2. Gain a broad understanding of Flask which is used to create a meaningful User Interface and Store the user's data efficiently and effectively.

2. LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

When you write a literature review in respect of your project, you have to write the researches made by various analysts - their methodology (which is basically their abstract) and the conclusions they have arrived at. You should also give an account of how this research has influenced your thesis.

Descriptive papers may or may not contain reviews, but analytical papers will contain reviews. A literature review must contain at least 5 - 7 published researches in your field of interest.

2.1 Existing problem

Ryan Ko et al.(2012).A study of mobile news reading suggests that users read the news once a day for between 10 and 30 minutes, preferably during the mornings and at home. The personalization of news app interaction is achieved through making the interface adaptable. Adaptive news interfaces that 'automatically' adapt to the way the user reads the news in particular contexts are not found, other than in re-ordering menus of headlines to take account of previous reading choices.

Marios constantinides et al,(2015).Mobile app ecosystems are transforming patterns of news help users keep abreast of news by aggregation over The personalization of news app interaction in these Adaptive news interfaces that 'automatically' adapt to the way the user reads the news in particular they jump to a particular section whereas when they read Having characterized mobile news readers as one of three types, the question that follows is whether

a news app could detect a user as being a particular reader type from their question involving the development of a mobile news app the adaptive variant interface for each news reader type.

Manish Kumar et al,(2016).In late 10 years, Internet has been growing appallingly rapidly. The expense of capacity, the force devoured by PC and equipment is expanding. Storage room in information focus can't address our needs and hence the framework and repair of unique web can't settle above enquiries, so new arrangements are required. In the meantime, substantial undertakings need to study information supply completely to bolster its business. The gathering and examination ought to be composed on a fresh out of the plastic new stage. So we require a fresh out of the plastic new registering model to use the empty assets of tablet, expand the financial productivity through up usage rate, diminish the hardware vitality utilization. This is an application software for daily user.in which a user gets news update from another user. This application serves best for each individual person.

2.2 References

[1] "Marios constantinides , Exploring mobile news reading interactions for news app personalization", 2015.

[2]"Ryan Ko, Markus Kirchberg, Bu Sung Lee, Tracking of Data Leaving the Cloud",2012.

2.3 Problem Statement Definition

Cloud Computing is one of the most widely used concepts around the world. It will be essential in storage of data. In this article, we will be dealing with news application which shows news according to user favorites. This produce good quality of news which is gathered from trusted source, Its also time consuming for the users.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

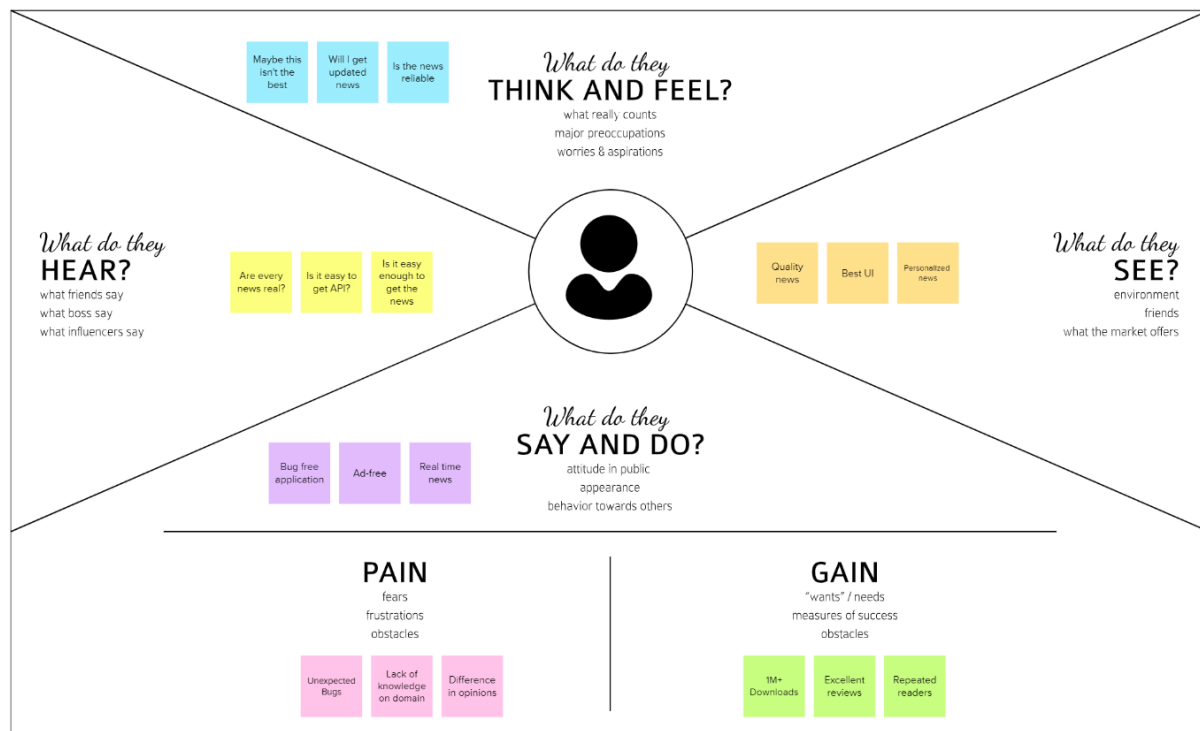


Fig. 3.1 EMPATHY MAP

3.2 Ideation & Brainstorming

Ideation refers to the whole creative process of coming up with and communicating new ideas. It can take many different forms, from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system. Ideation is similar to a practice known as brainstorming.

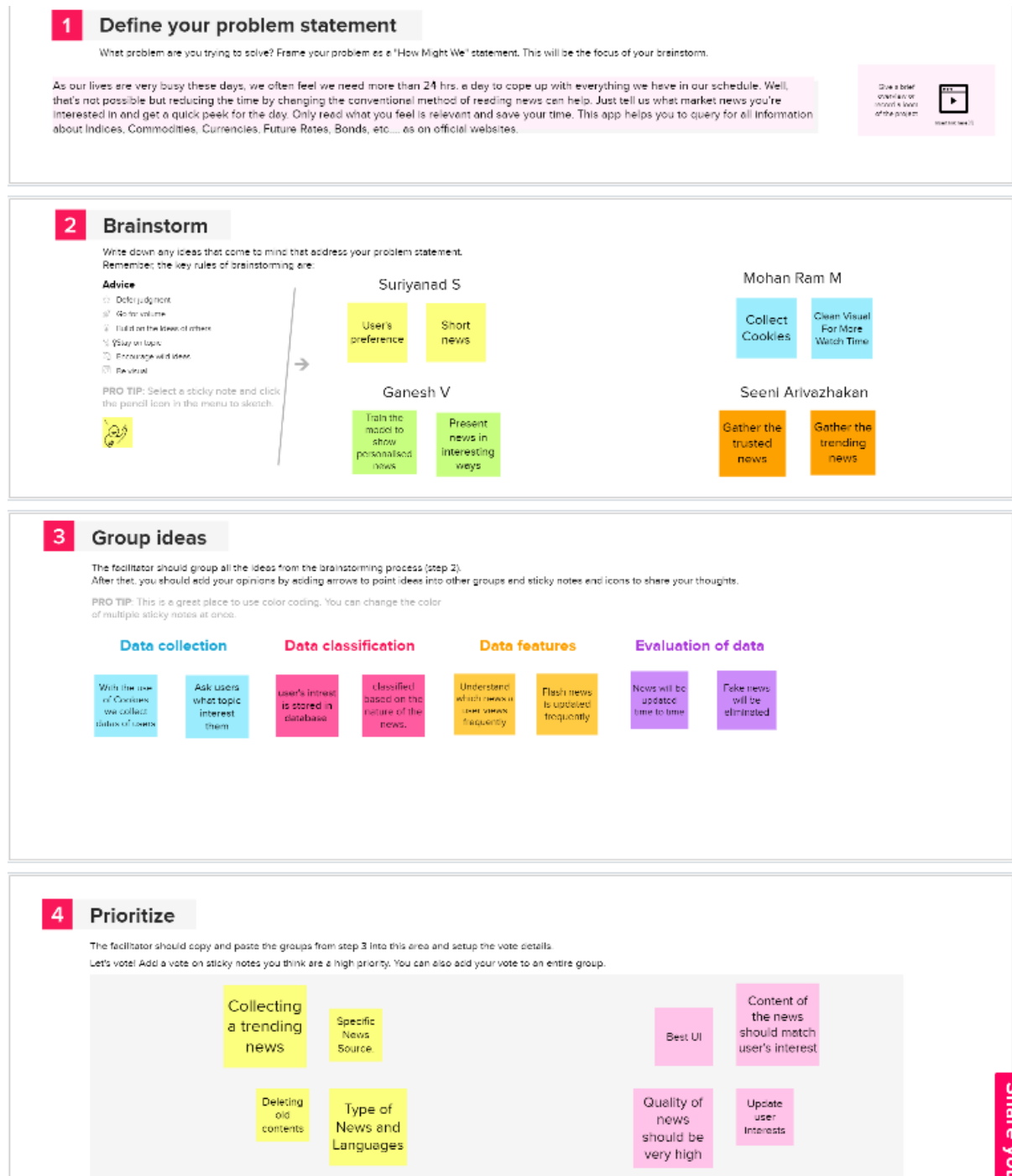


Fig. 3.2 BRAINSTORMING

3.3 Proposed Solution

Proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

Table 3.1 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Cloud Computing is one of the fast growing and it offers flexible resources, and economies of scale. In this article we are going make an application which tracks the news. It is an News Tracker Application where it gives crisp news from the reputed news sources and it's just gives the simple sentence which saves the readers time and also it is high in knowledge.
2.	Idea / Solution description	We are using html - bootstrap ,python - flask. This is for the continuous display ofnews. As Python has lot of libraries ,it is easier and faster.
3.	Novelty / Uniqueness	Usually people use javascript ,php as a backend for web development. But we are using flask for that purpose.
4.	Social Impact / Customer Satisfaction	Less time and more knowledge consumption.
5.	Business Model (Revenue Model)	Optional(Ad-revenue)
6.	Scalability of the Solution	Design,news and flash news are flexible.

3.4 Problem Solution fit

Project Title: NEWS Tracker Application

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID32880

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Working individuals and passionate NEWS readers	6. CUSTOMER CONSTRAINTS CC For ad free interface every news provider demand subscription from users	5. AVAILABLE SOLUTIONS AS Used Adblockers to avoid ad, but it's not secure for the user	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Clumsy UI, Not Personalized, Vague news	9. PROBLEM ROOT CAUSE RC Now a days news has become a money making tool for the news agency so they force customer to pay for subscription or use ad loaded free version	7. BEHAVIOUR BE Pay for subscription or adjust with the ads shown	
Focus on J&P, tap into BE, understand RC				Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TR Seeing others using NEWS app with ad free interface motivate users to download the same	10. YOUR SOLUTION SL Provide free access to ad-free UI to read news, by using free APIs available to Provide personalized news to the users	8. CHANNELS of BEHAVIOR CH <u>ONLINE</u> Uses memes and posts in social media to get updated about happenings <u>OFFLINE</u> Talk with peoples to know the current events and happenings	Identify strong TR & EM

Fig. 3.3 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. Generally functional requirements describe system behavior under specific conditions.

Table 4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	<ul style="list-style-type: none">Registration through FormRegistration through Gmail
FR-2	User Confirmation	<ul style="list-style-type: none">Confirmation via EmailConfirmation via OTP
FR-3	Searching	<ul style="list-style-type: none">Search the news based on the user interest.Trending news will be shown.User can choose the genre.
FR-4	Real-time News	<ul style="list-style-type: none">User could see the real news from trusted sources.Auto update news because it fetch news from NEWS-API.
FR-6	Location Based	<ul style="list-style-type: none">User could see their country news.They could also see other country news.

4.2 Non-functional requirement

Nonfunctional requirements, not related to the system functionality, rather define how the system should perform.

Table 4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none">The news is updated almost every second, it does not affect the user who reads his/her favourite news.
NFR-2	Security	<ul style="list-style-type: none">Authentication and Password Management.
NFR-3	Reliability	<ul style="list-style-type: none">Avoid fake news.Instant news.Location based news.
NFR-4	Performance	<ul style="list-style-type: none">Smooth UI makes the user to stay in app.Showing the current trending news first.
NFR-5	Availability	<ul style="list-style-type: none">Minimum idle time is 24/7.Recent news will be based on user's previous news visit.
NFR-6	Scalability	<ul style="list-style-type: none">Get more user's attention by sharing the app in social media

5. PROJECT DESIGN

5.1 Data Flow Diagram

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 can be manual, automated, or a combination of both.

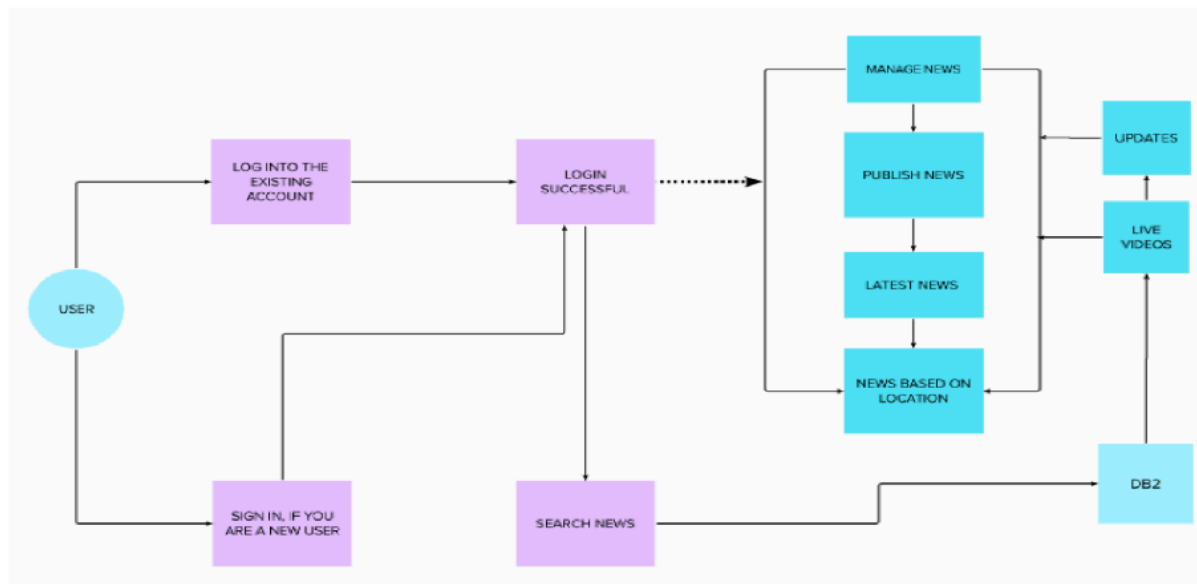


Fig. 5.1 DATAFLOW DIAGRAM

5.2 Solution & Technical Architecture

IBM® Cognos® Analytics with Watson takes BI a step further with AI capabilities that not only bring an accurate, trusted and complete picture of your business, but forecast what's coming in the future, predict outcomes and explain why they may happen.

Why IBM Cognos Analytics with Watson?

- Use built-in AI to accelerate and improve blending data or finding the optimal tables for your model.
- Uncover hidden trends and drivers with the help of AI so you can get the facts behind your data and deliver insights in real time.

- Create powerful visualizations, tell the story of your data and share insights via email, Slack, or the mobile app.

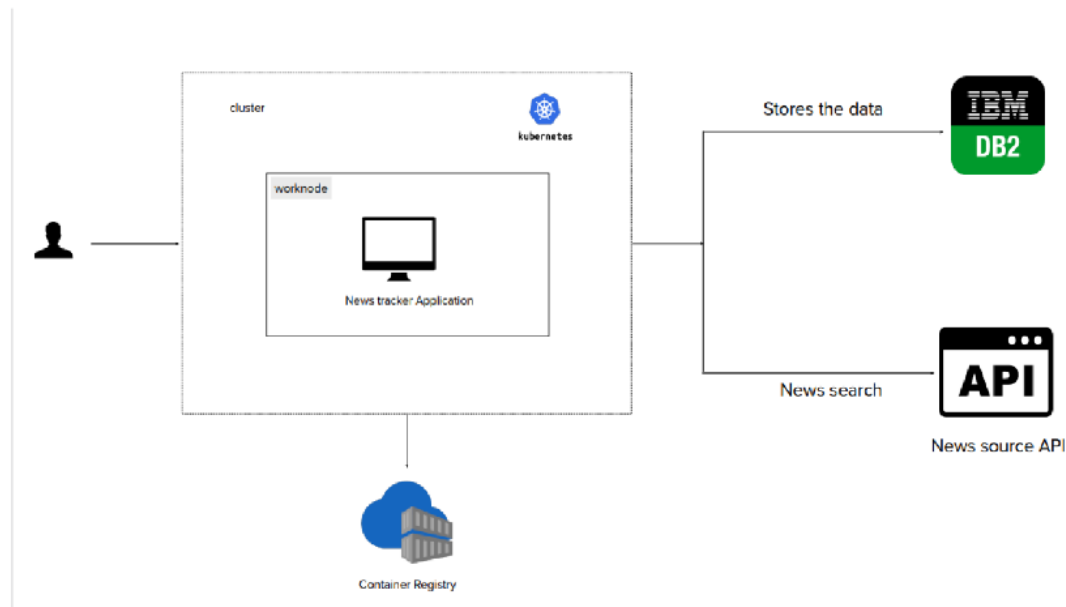


Fig. 5.2 TECHNICAL ARCHITECTURE

5.3 User Stories

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

Table 5.1 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can start reading news.	High	Sprint-1
	Dashboard	USN-3	The news portal fetches for the most recent news and shows it as "Breaking News".	I can open and view the "Breaking News"	Medium	Sprint-2
	Search Bar	USN-4	User search data based on their personal interests	I can view the related content for the search data	High	Sprint-2
		USN-5	The news is showed with a sample snapshot and mini description and user can open and read the article.	I can view the news details and can open to see whole article	Medium	Sprint-3
Administrator	Server	USN-6	Provides the exact news from the database	I can see the latest news getting refreshed	Medium	Sprint-4
		USN-7	Provides Personalized News based on the interest	I can see the news based on my interest	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Table 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.	2	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	3	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint-2	Dashboard	USN-5	As a user, I can search a news and a snippet will displayed on the screen	5	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Database	USN-4	Connection to the database for maintaining the user details.	3	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint-3	Notifications	USN-4	As a user, I will receive notifications to my email so that I'll be updated on the news	2	Medium	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.
Sprint-4	Monitor the system	USN-5	As a admin I should be able to monitor the cloud system and fix the errors before customer	1	High	Ganesh V, Mohan Ram M, Seeni Arivazhakan, Suriyanad S.

6.2 Sprint Delivery Schedule

Table 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	8	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	4	19 Nov 2022

6.3 Burn down chart

Sprint no.	Dates		Story Points
	Start	End	
1	Oct-24	Oct-29	8
2	Oct-31	Nov-05	4
3	Nov-07	Nov-12	5
4	Nov-14	Nov-19	4

BURNDOWN CHART

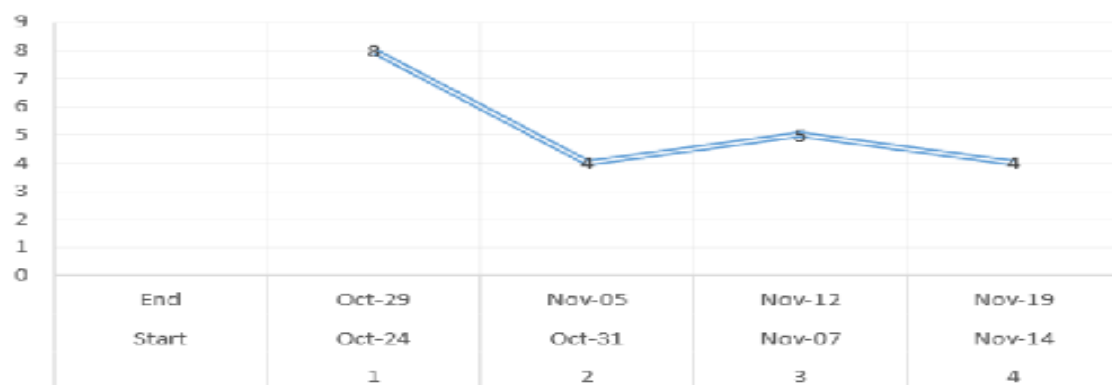


Fig. 6.1 BURN DOWN CHART

7. TESTING

Testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

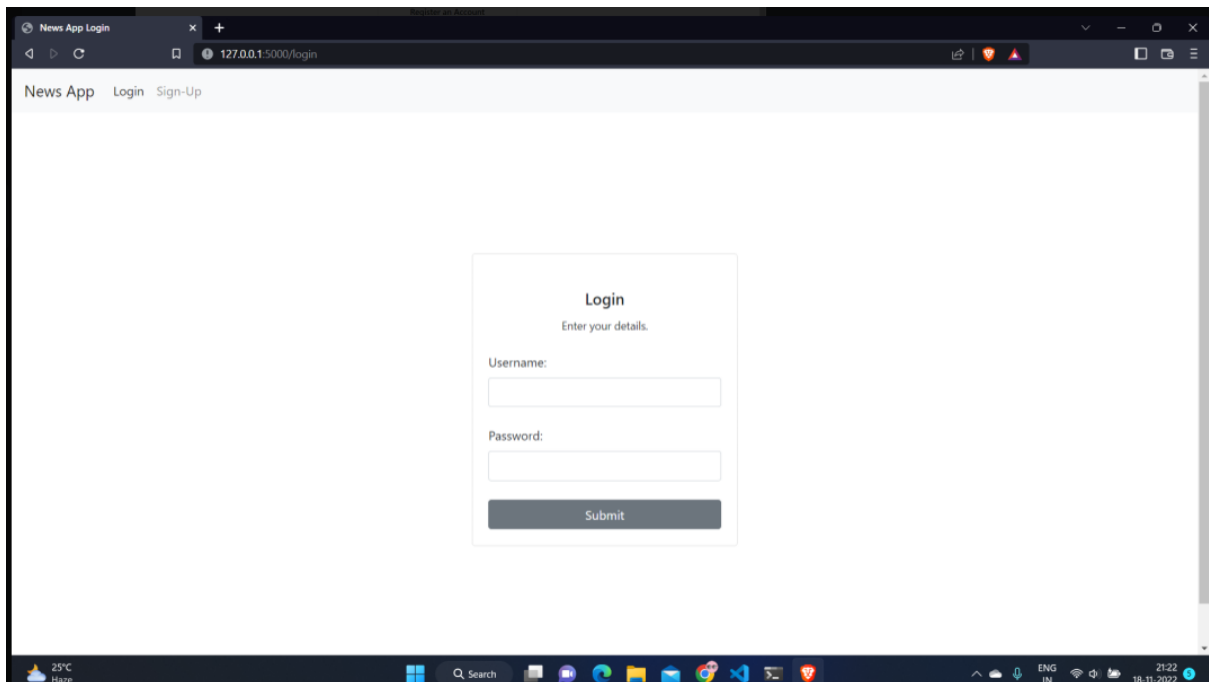
7.1 Test Cases

Table 7.1 Test Cases

S.NO	TESTCASE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1.	User login	Username& Password	Successful login to home page	Successful login to home page	PASS
2.	Storing data	Registered credentials	Data is stored in Database	Data is stored in Database	PASS
3.	Display news	Login the website	New news displayed	New news displayed	PASS

8. OUTPUT

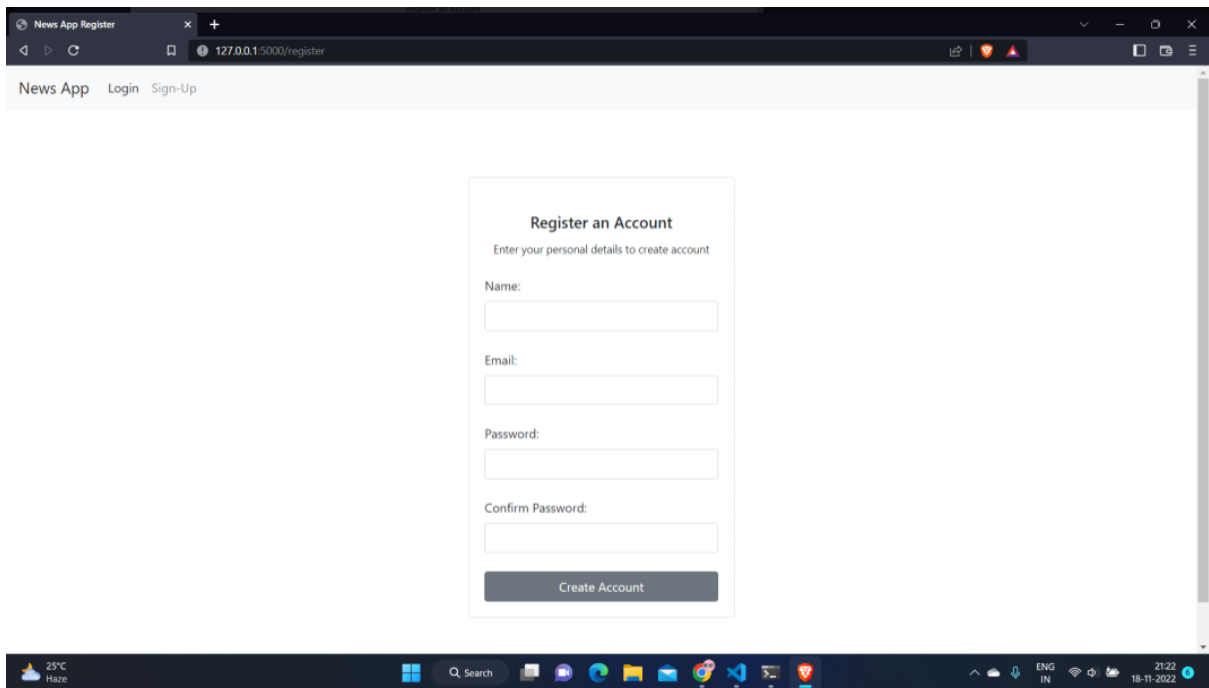
8.1 Login page



The screenshot shows a web browser window titled "News App Login" with the address bar displaying "127.0.0.1:5000/login". The page has a navigation bar with "News App", "Login", and "Sign-Up" links. The main content area features a "Login" form with the subtitle "Enter your details.". The form includes a "Username:" label with a text input field, a "Password:" label with a text input field, and a "Submit" button at the bottom.

Fig. 8.1 LOGIN PAGE

8.2 Register page



The screenshot shows a web browser window titled "News App Register" with the address bar displaying "127.0.0.1:5000/register". The page has a navigation bar with "News App", "Login", and "Sign-Up" links. The main content area features a "Register an Account" form with the subtitle "Enter your personal details to create account". The form includes a "Name:" label with a text input field, an "Email:" label with a text input field, a "Password:" label with a text input field, and a "Confirm Password:" label with a text input field. A "Create Account" button is located at the bottom of the form.

Fig. 8.2 REGISTER PAGE

8.3 Home page

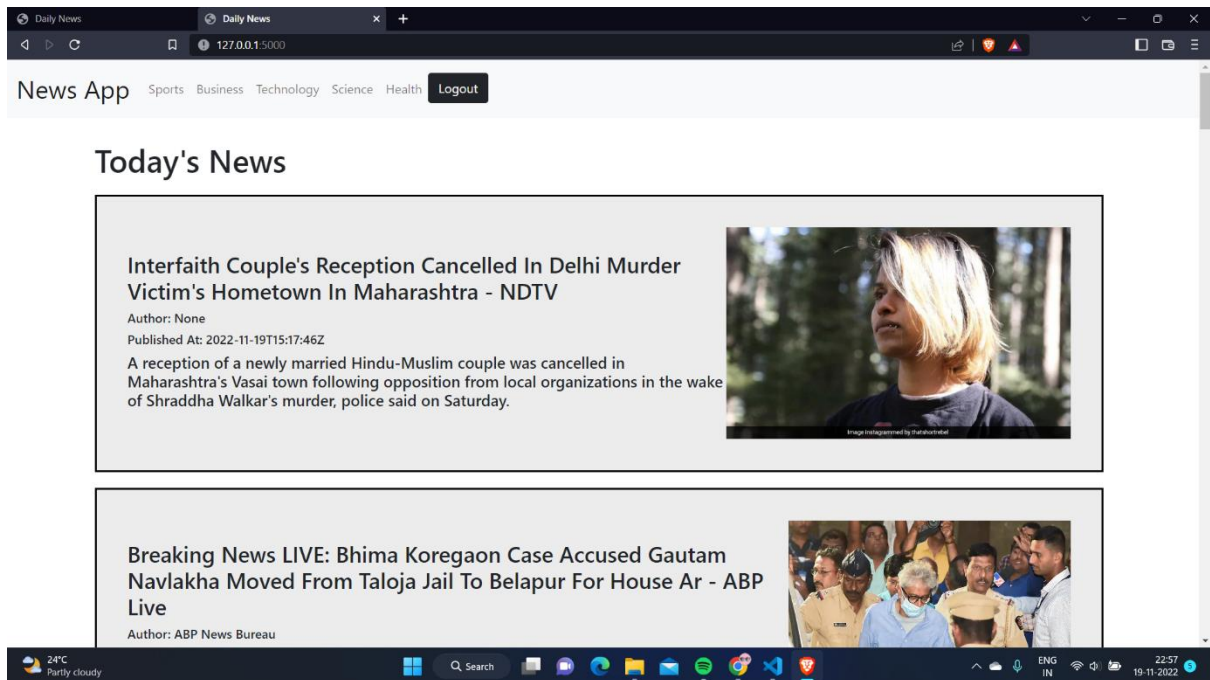


Fig. 8.3 TODAY'S NEWS

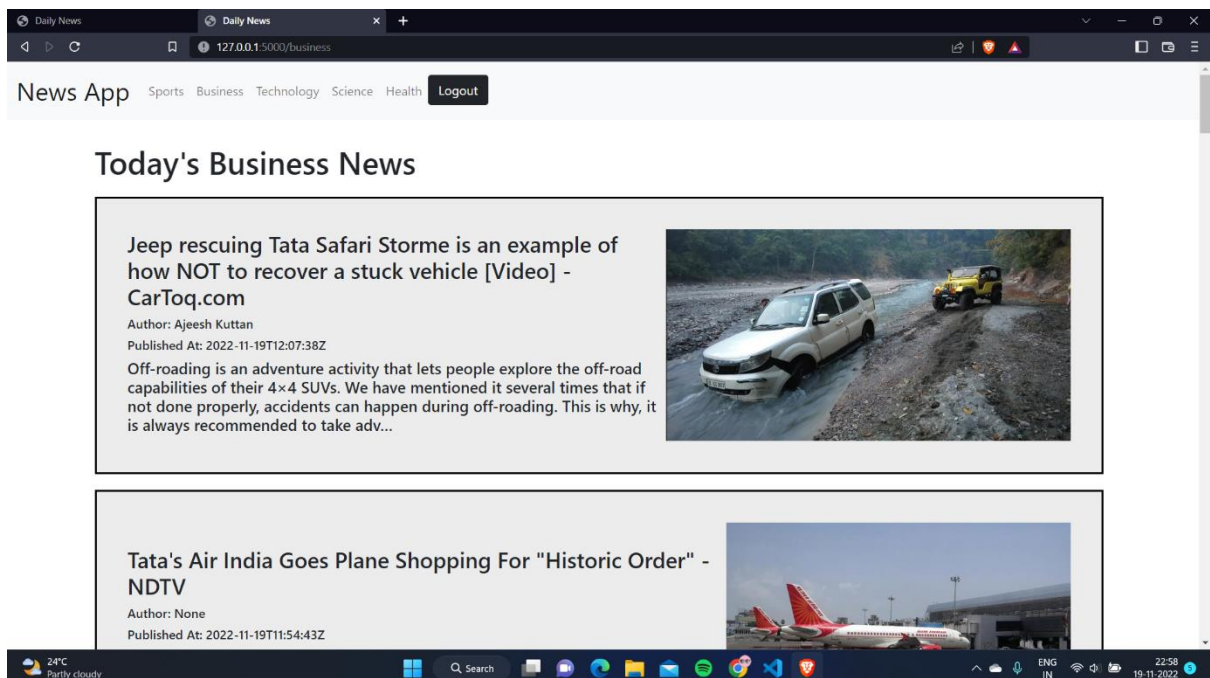


Fig. 8.4 BUISNESS NEWS

The screenshot displays the IBM Db2 on Cloud management console. The 'Connections' tab is selected, showing a table of active connections. The table has columns for Application name, Client user ID, Client IP address, Application handle, Connection start time, Workload name, Service superclass name, and Workload occurrence. A single connection for 'python.exe' is visible.

Application name	Client user ID	Client IP address	Application handle	Connection start time	Workload name	Service superclass name	Workload occurrence
python.exe		172.30.68.128	21251	Nov 24, 2022 9:31:27 PM	SYSDEFAULTUSER WORKLOAD	SYSDEFAULTUSERCLASS	UOWWAIT

Fig.8.5 DATABASE CONNECTION.

9. ADVANTAGES & DISADVANTAGES

Advantages:

- Improved availability of data. DB2 data sharing helps you meet your service objective by improving availability during both planned and unplanned outages.
- Faster time to market. You can spin up new instances or retire them in seconds, allowing developers to accelerate development with quick deployments.
- flask is lightweight, open source, and offer minimal coding for developing an application.

Disadvantages:

- More potential for security risks.
- IBM Cloud with a bare metal server is not the best solution for you as the configuration of this server is usually a rather time-consuming process.
- The complete plan must be rebound for any DB2 program modification.
- BIND can be time-consuming for large plans.

10. CONCLUSION

Thus a news tracker application was created using Flask, IBM DB2 and the data's are stored in IBM cloud which displays the current news when logged in with right credentials

11. FUTURE SCOPE

User interest based news and most interested news can be displayed accurately with the help of Machine learning which can be implemented in the future

12. APPENDIX

SOURCE CODE

app.py:

```
from flask import Flask, render_template,request,session,redirect
import requests
import ibm_db
import ibm_db_dbi
import re
from flask_db2 import DB2

app = Flask(__name__)
app.secret_key = 'secret123'

app.config['database'] = 'bludb'
app.config['hostname'] = '55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud'
app.config['port'] = '31929'
app.config['protocol'] = 'tcpip'
app.config['uid'] = 'ntj24419'
app.config['pwd'] = 'fTbavf47TXpOwDUw'
app.config['security'] = 'SSL'
try:
```

```

mysql = DB2(app)

conn_str='database=bludb;hostname=55fbc997-9266-4331-afd3-
888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;port=31929;protocol=tcpi
p;uid=ntj24419;pwd=fTbavf47TXpOwDUw;security=SSL'
ibm_db_conn = ibm_db.connect(conn_str,"")

print("Database connected without any error !!")
except:
    print("IBM DB Connection error : " + ibm_db.conn_errormsg())

@app.route('/')
def home():
    #url = "https://newsapi.org/v2/top-
    headlines?country=in&category=general&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    #r= requests.get(url).json()
    #case = {
    #    'articles': r['articles']
    #}
    return render_template('login.html')#cases = case)

@app.route('/index')
def index():
    url = "https://newsapi.org/v2/top-
    headlines?country=in&category=general&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('index.html',cases=case)

@app.route('/sports')
def sports():
    url = "https://newsapi.org/v2/top-
    headlines?country=in&category=sports&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }

```

```

return render_template('sports.html',cases = case)

@app.route('/business')
def business():
    url = "https://newsapi.org/v2/top-
headlines?country=in&category=business&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('business.html',cases = case)

@app.route('/technology')
def technology():
    url = "https://newsapi.org/v2/top-
headlines?country=in&category=technology&apiKey=7e1737d3191d4fe894fc579df01b7bde
"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('tech.html',cases = case)

@app.route('/science')
def science():
    url = "https://newsapi.org/v2/top-
headlines?country=in&category=science&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('science.html',cases = case)

@app.route('/health')
def health():
    url = "https://newsapi.org/v2/top-
headlines?country=in&category=health&apiKey=7e1737d3191d4fe894fc579df01b7bde"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']

```

```

    }
    return render_template('health.html',cases = case)

@app.route('/reg')
def reg():
    return render_template('register.html')

@app.route('/register',methods=["GET","POST"])
def register():
    username = request.form.get("name")
    email = request.form.get("email")
    password = request.form.get("password")

    try:
        connectionID = ibm_db_dbi.connect(conn_str, ", ")
        cursor = connectionID.cursor()
    except:
        print("No connection Established")

    sql = "SELECT * FROM register WHERE username = ?"
    stmt = ibm_db.prepare(ibm_db_conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt)
    result = ibm_db.execute(stmt)
    print(result)
    account = ibm_db.fetch_row(stmt)
    print(account)

    param = "SELECT * FROM register WHERE username = " + "\"" + username + "\""
    res = ibm_db.exec_immediate(ibm_db_conn, param)
    print("---- ")
    dictionary = ibm_db.fetch_assoc(res)
    while dictionary != False:
        print("The ID is : ", dictionary["USERNAME"])
        dictionary = ibm_db.fetch_assoc(res)

    if account:
        msg = 'Username already exists !'
    elif not re.match(r'^[@]+@[^@]+\.[^@]+$', email):
        msg = 'Invalid email address !'

```

```

elif not re.match(r'[A-Za-z0-9]+', username):
    msg = 'name must contain only characters and numbers !'
else:
    sql2 = "INSERT INTO register (username, email,password) VALUES (?, ?, ?)"
    stmt2 = ibm_db.prepare(ibm_db_conn, sql2)
    ibm_db.bind_param(stmt2, 1, username)
    ibm_db.bind_param(stmt2, 2, email)
    ibm_db.bind_param(stmt2, 3, password)
    ibm_db.execute(stmt2)
    msg = 'You have successfully registered !'
    print(msg)

return render_template('login.html')

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/loginn',methods =['GET', 'POST'])
def loginn():
    title = "Home"
    if request.method == 'POST' :
        username = request.form['uname']
        password = request.form['password']

        sql = "SELECT * FROM register WHERE username = ? and password = ?"
        stmt = ibm_db.prepare(ibm_db_conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        result = ibm_db.execute(stmt)
        print(result)
        account = ibm_db.fetch_row(stmt)
        print(account)

        param = "SELECT * FROM register WHERE username = " + "\"" + username + "\"" + "
and password = " + "\"" + password + "\""
        res = ibm_db.exec_immediate(ibm_db_conn, param)
        dictionary = ibm_db.fetch_assoc(res)

```

```

if account:
    session['loggedin'] = True
    session['username'] = dictionary["USERNAME"]
    session['email'] = dictionary["EMAIL"]

    #return render_template('base.html')
    return redirect('/index')
else:
    return render_template('login.html')

```

```

return render_template('login.html')

```

```

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

```

base.html:

```

<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <title>Daily News</title>
    <style>
        .container {
            margin-top: 20px;
        }
        h1{
            margin-top: 8%;
        }
        .nav{
            overflow: hidden;
            position: fixed;

```

```

top: 0;
width: 100%;
}
.news_box {
border: 3px solid rgb(0, 0, 0);
margin-top: 20px;
margin-bottom: 20px;
margin-right: 10px;
padding: 3%;
display: flex;
align-items: center;
justify-content: space-between;
font-size: 1.2em;
background-color: rgb(236, 236, 236);
}
.navbar-brand {
text-decoration: none;
font-size: 2em;
}

@media screen and (max-width: 980px) {
.news_box {
flex-direction: column;
}
.img_div img{
width: 100%;
}
}
</style>
</head>
<body>
<nav class="navbar nav navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="{ { url_for('index') } }">News App</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">

```

```

<ul class="navbar-nav">
  <li class="nav-item">
    <a class="nav-link" href="{{ url_for('sports') }}">Sports</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{ url_for('business') }}">Business</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{ url_for('technology') }}">Technology</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{ url_for('science') }}">Science</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{ url_for('health') }}">Health</a>
  </li>
  <li class="nav-item">
    <a href="{{ url_for('login') }}" class="btn btn-dark float-end">Logout</a>
  </li>
</ul>
</div>
</div>
</nav>
{% block body %}

{% endblock body %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
</html>

```


index.html:

```
{% extends 'base.html' %}
{% block body %}
<div class="container">
  <h1>Today's News</h1>
  {% for news in cases.articles %}
  <div class="news_box">
    <div class="info_div">
      <h3 class="title">{{ news.title }}</h3>
      <h6>Author: {{ news.author }}</h6>
      <h6>Published At: {{ news.publishedAt }}</h6>
      <h5>{{ news.description }}</h5>
    </div>
    <div class="img_div">
      
    </div>
  </div>
  {% endfor %}
</div>
{% endblock body %}
```

Project GitHub Link :

<https://github.com/IBM-EPBL/IBM-Project-16312-1659611345>

Project Demo Link :

<https://github.com/IBM-EPBL/IBM-Project-16312-1659611345/tree/main/Final%20Deliverables/Demo%20Video>