# NEWS TRACKER APPPLICATION

USING CLOUD

*A Project report submitted in partial fulfilment of 7th semester in degree of*

BACHELOR OF ENGINEERING

IN

# COMPUTER SCIENCE ENGINEERING

***Submitted by***

Team ID: PNT2022TMID12589

19z229 Navaneeth A.B.

19z257 Venigalla Akhil

19z262 yogesh kumaar.R

20z433 Srinath.S

**ABSTRACT**

Everyone has the right to freedom of speech. However, this right is being misused to differentiate and attack others, physically or verbally, in the name of free speech. This discrimination is known as hate speech. Hate speech can be well-defined as language used to express hate towards a person or a group of people based on characteristics such as race, religion, ethnicity, gender, nationality, disability and sexual orientation. The increasing usage of social sites and information sharing has specified major benefits to humanity. However, this has also assumed rise to a variety of challenges including the spreading and sharing of hate speech messages. Thus, to solve this emerging issue in social media sites, recent studies employed a variety of machine learning and deep learning algorithms with text mining algorithm to automatically detect the hate speech messages on real time datasets. Hence, the aim of this Project is to analyses the comments on social networks using Natural Language processing technique (NLP) and Deep learning algorithm named as Back propagation neural network algorithm. Using NLP technique, can extract the keywords from user generated content and implement Back Propagation neural network to classify the text whether it is positive or negative. If it is negative means, automatically block the comments as per user wish and also block the friends based on pre-defined threshold values. Experimental results shows that the proposed framework implemented in real time social network site with improved notification .

**TABLE OF FIGURES**

1. INTRODUCTION
   1. Project Overview
   2. Purpose
2. LITERATURE SURVEY 2.1 Existing problem
   1. References
   2. Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas
   1. Ideation & Brainstorming
   2. Proposed Solution
   3. Problem Solution fit
4. REQUIREMENT ANALYSIS 4.1 Functional requirement
   1. Non-Functional requirements
5. PROJECT DESIGN

1. PROJECT PLANNING & SCHEDULING 6.1 Sprint Planning & Estimation
   1. Sprint Delivery Schedule
   2. Reports from JIRA
2. CODING & SOLUTIONING

1. ADVANTAGES & DISADVANTAGES
2. CONCLUSION
3. FUTURE SCOPE
4. APPENDIX

Source Code GitHub & Project Demo Link

# News Tracker Application

## 1.INTRODUCTION

Mobile app ecosystems are transforming patterns of news consumption. Until quite recently, reading the news was a niche use for smartphones [12], mostly for when users were ‘on the go’; now however, two in every three users of mobile devices in the US regularly access news and as many as one in five read in-depth news articles daily [2]; a similar picture is found in the UK [1]. This growth in mobile news access continues the migration of news consumers to the Internet.

### 1.1 Project overview

As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces. We first surveyed users’ news reading preferences and behaviours; analysis revealed three primary types of reader. We then implemented and deployed an Android news app that logs users’ interactions with the app. We used the logs to train a classifier and showed that it is able to reliably recognise a user according to their reader type. Finally we evaluated alternative, adaptive user interfaces for each reader type. The evaluation demonstrates the differential benefit of the adaptation for different users of the news app and the feasibility of adaptive interfaces for news apps.

### 1.2 Purpose

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.

### 2. LITERATURE SURVEY 2.1 Existing problem

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.

#### 2.2 References

Like articles, websites, blogs.

https://www.researchgate.net/.

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

**https://www.ripublication.com/**

#### 2.3 Problem Statement Definition

1.The user needs a way to organize the news on daily basis so that he can cope up with daily events in his tight schedule.

2.The user wants to read news only about particular topics so that he can be informed about his interest.

3.The user wants to get informed from only certified news outlets .

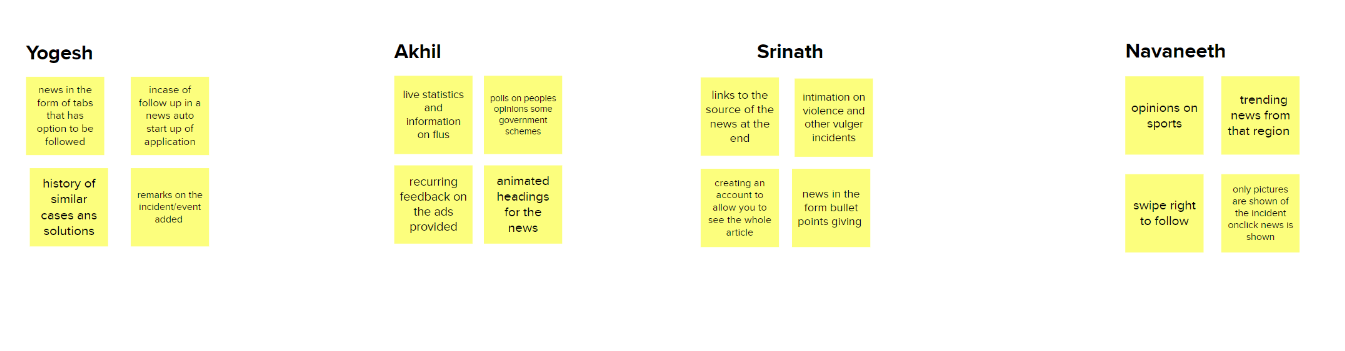
4.The user needs a way to search about the news on topics he wants to.

5.The user wants to know the news about his surrounding using GPS location.

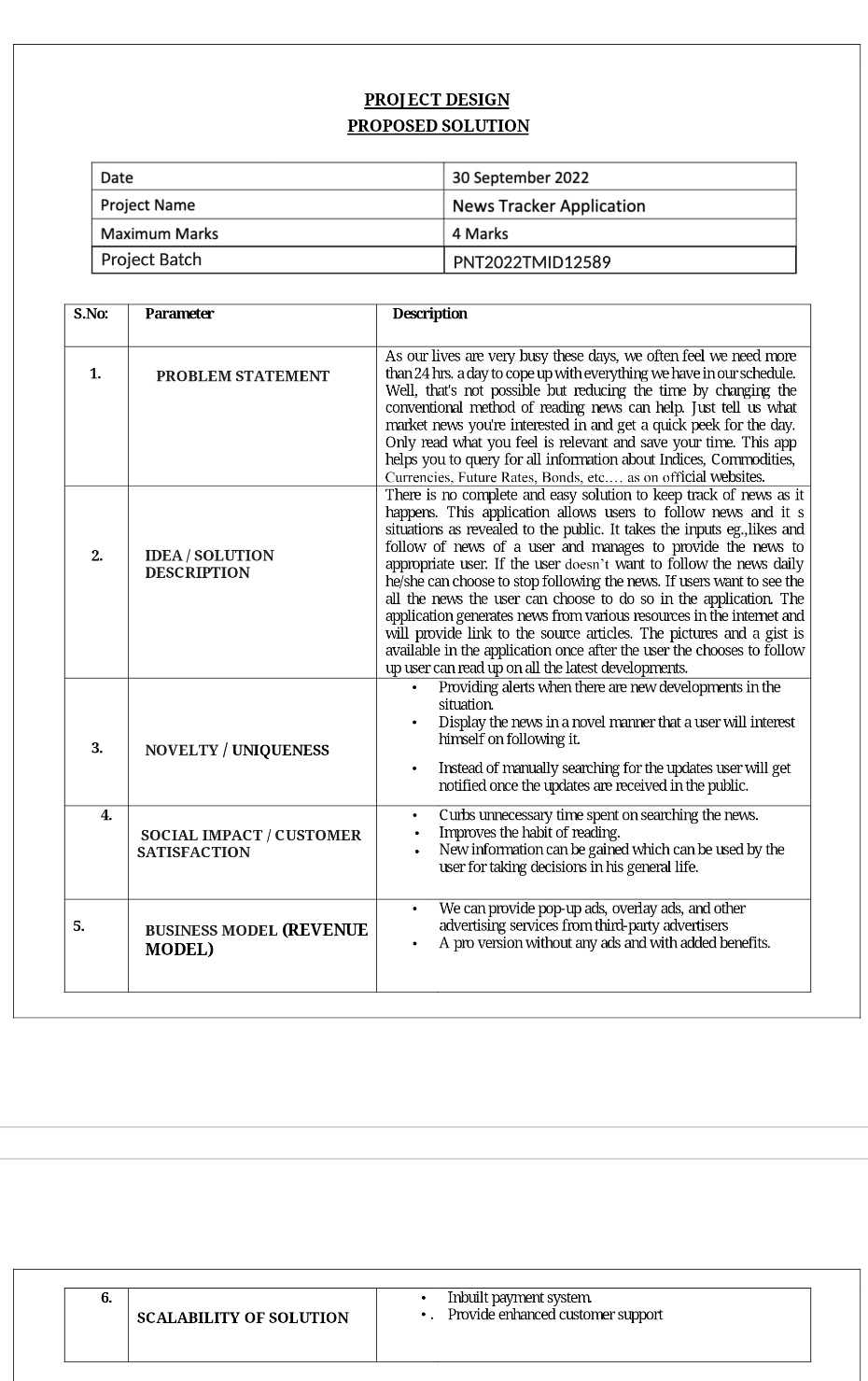
### 3. IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas

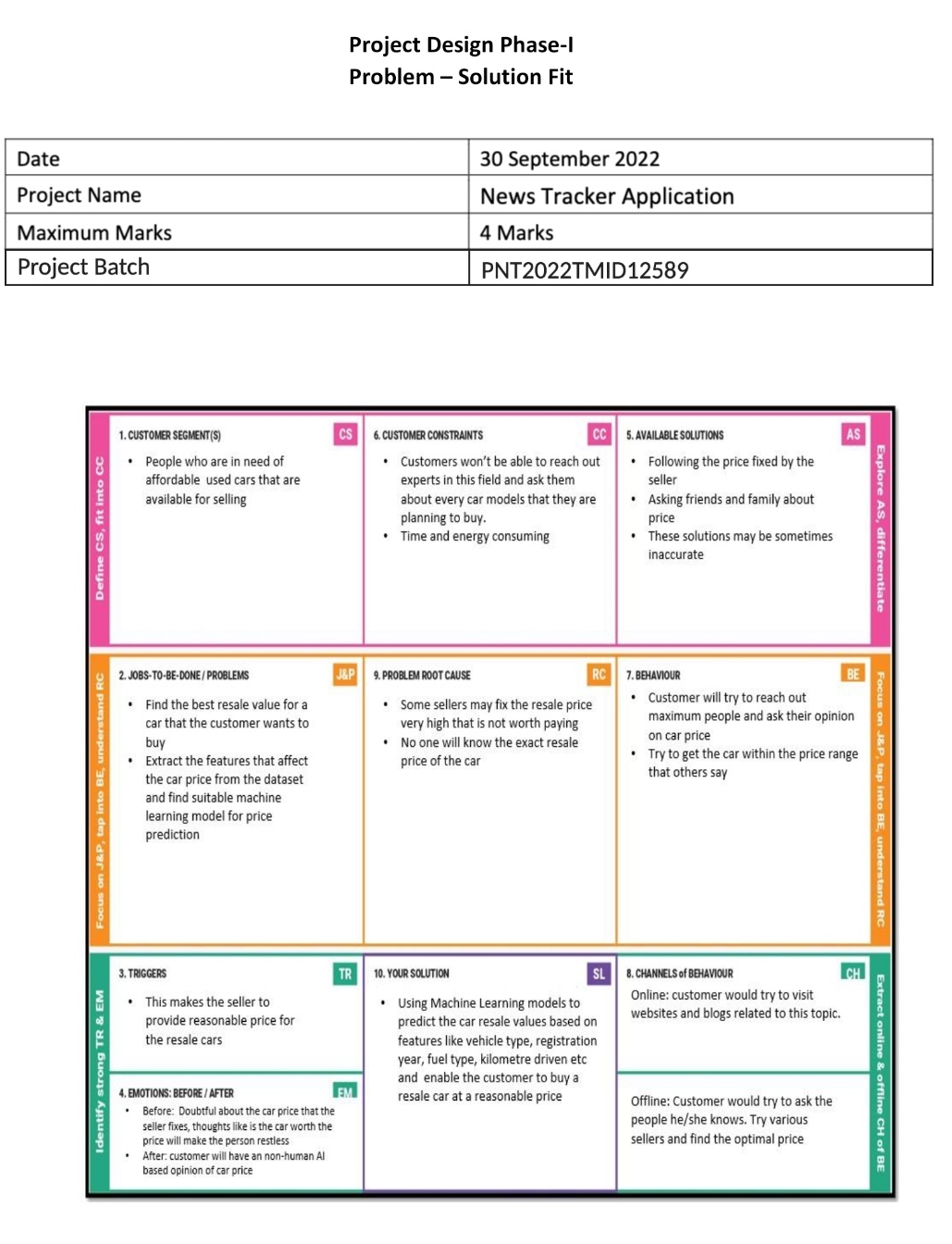


#### 3.2 Ideation & Brainstorming



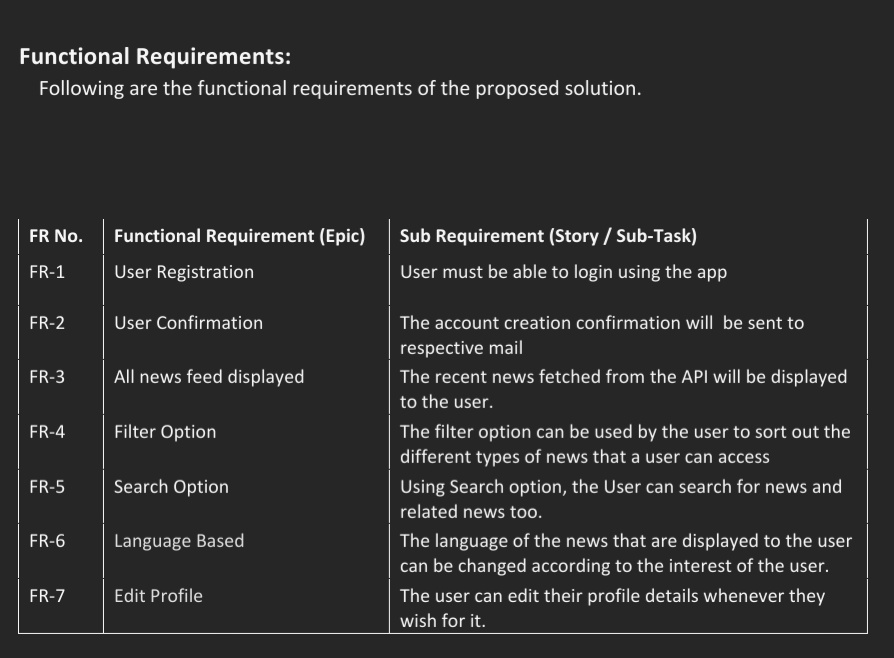
3.3 Proposed Solution



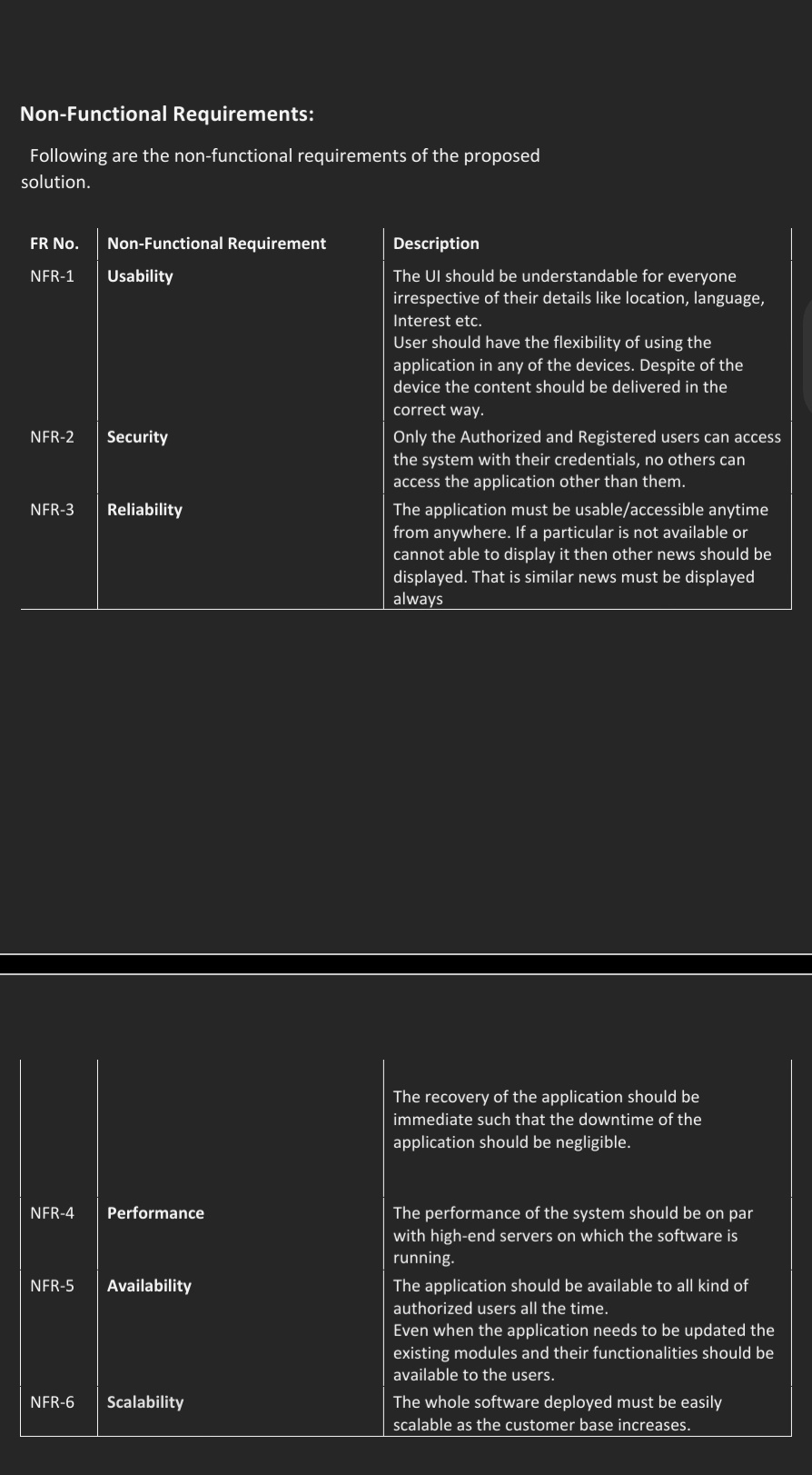
3.4 Problem Solution fit

### 4. REQUIREMENT ANALYSIS

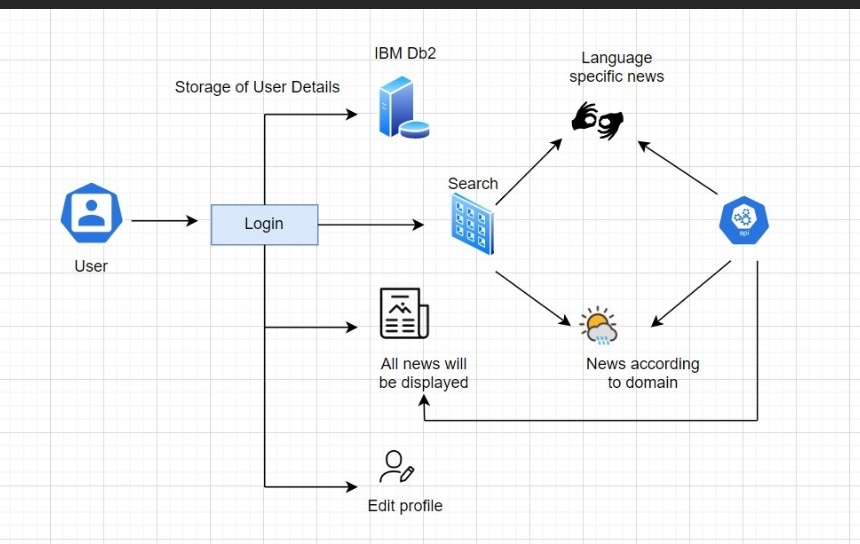
4.1)



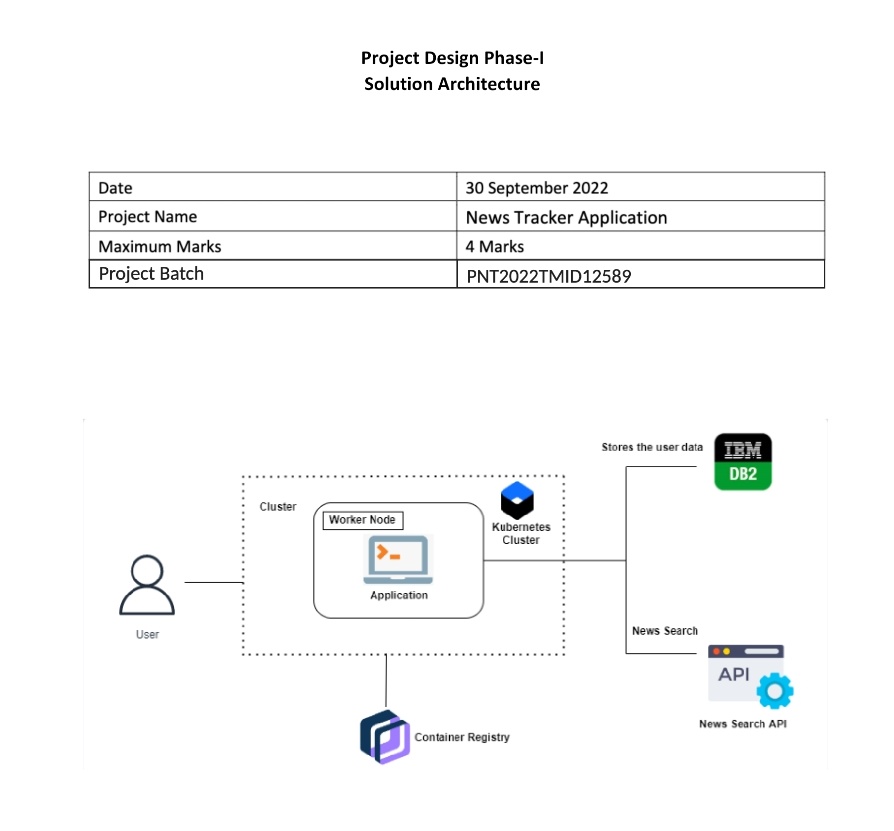
#### 4.2)



### 5. PROJECT DESIGN 5.1 Data Flow Diagrams



#### 5.2 Solution & Technical Architecture



##### 6. PROJECT PLANNING & SCHEDULING

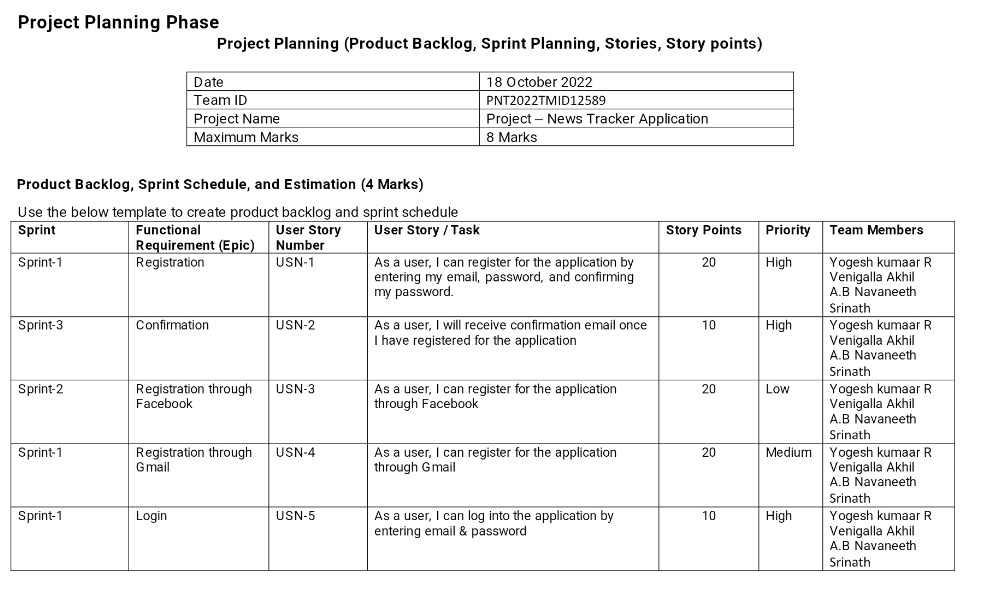
##### 6.1 Sprint Planning & Estimation

|  |  |  |
| --- | --- | --- |
| TITLE | DESCRIPTION | DAT  E |
| Literature Survey & | Literature survey on the |  |
| Information Gathering | selected project & gathering information by referring the, technical papers, research publications etc. | 2 Sept 2022 |
| Prepare Empathy Map | Prepare Empathy Map  Canvas to capture the user Pains & Gains, Prepare list of problem statements | 10 Sept 2022 |

|  |  |  |
| --- | --- | --- |
| Ideation | List the by organizing the |  |
| Brain Storming | brainstorming session and prioritize the top 3 ideas based on the feasibility & importance. | 15 Sept 2022 |
| Proposed Solution | Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc. | 23 Sept 2022 |
| Problem Solution Fit | Prepare problem solution fit document | 28 Sept 2022 |

|  |  |  |
| --- | --- | --- |
|  | understand the user interactions  & experiences with the application (entry toexit). |  |
| Solution Requirement | Prepare the functional requirement document | 8 October 2022 |
| Data Flow Diagrams | Draw the data flow diagrams and submit for review. | 10 October 2022 |
| Technology | Prepare the technology |  |
| Architecture | architecture diagram. | 15 October 2022 |
| Prepare Milestone & | Prepare the milestones & | 26 October 2022 |
| Activity  List | activity list of the project. |  |
| Project Development - | Develop & submit the |  |
| Delivery of Sprint-1, 2, 3  & 4 | developed code by testing it. | IN PROGRESS.. |

#### 6.2 Sprint Delivery Schedule



**6.3 Reports from JIRA**

## ------------------------------------------------------------------------------------------------ ----------------

### 7. CODING & SOLUTIONING

SOURCE CODE:

TEMPLATES:

LOGIN PAGE:

<html>

<head>

<title>LOGIN</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta charset="utf-8" />

<link rel="stylesheet" type="text/css"

href="{{ url\_for('static', filename='pages.css') }}"

/>

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/fontawesome.min.css"

/>

<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottieplayer.js"></script>

<link rel="stylesheet"

href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-

vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp" crossorigin="anonymous"

/>

</head>

<body class="body">

<div class="on-screen">

<div class="form">

<form action="/login" class="innercont" method="POST">

<br />

<button

style="width: 290px;margin-bottom: 24px;" type="button"

onclick="window.location.href='/signup'"

>

CREATE AN ACCOUNT</button

><br />

<input name="username"

type="text"

placeholder="&#xf007; Username" required

/>

<input name="password" type="password" id="password"

placeholder="&#xf023; Password" required

/><br />

<i class="fas fa-eye" onclick="show()"></i><br />

<h4>{{msg}}</h4>

<br />

<button style="width: 135px" type="submit">LOGIN</button>

<p class="message"></p>

</form>

</div>

</div>

<script> *function* show() {

*var* password = document.getElementById("password"); *var* icon = document.querySelector(".fas"); if (password.type === "password") { password.type = "text";

} else {

password.type = "password";

}

}

</script>

</body>

</html>

NEWACCOUNT:

<html>

<head>

<title>NEW ACCOUNT</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='pages.css') }}">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">

<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottieplayer.js"></script>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-

vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp" crossorigin="anonymous">

</head>

<body class="body">

<div class="on-screen">

<div class="form">

<form action="/login" method="POST">

<lottie-player src="https://assets2.lottiefiles.com/private\_files/lf30\_3ghvm6sn.json" speed="1" style="justify-content: center;" loop autoplay></lottie-player><br>

<center><h4>Your account has been successfully registered and the login credentials were shared with you to the registered email address.</h4><br><center>

<input name="username" type="text" placeholder="&#xf007; Username" required />

<input name="password" type="password" id="password" placeholder="&#xf023;

Password" required/><br>

<i class="fas fa-eye" onclick="show()"></i><br><br>

<button type="submit">LOGIN</button>

<p class="message"></p>

</form>

</div>

</div>

<script>

*function* show()

{

*var* password = document.getElementById("password"); *var* icon = document.querySelector(".fas")

if(password.type === "password")

{

password.type = "text";

} else

{

password.type = "password";

}

};

</script>

</body>

</html>

NEWS:

html>

<head>

<meta charset="UTF-8" />

<title>CAD-NEWSTRACKER</title>

<link

rel="stylesheet" type="text/css"

href="{{ url\_for('static', filename='news.css') }}"

/>

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottieplayer.js"></script>

<link

href="https://fonts.googleapis.com/css2?family=Cormorant+Garamond:ital,wght@1,700&d isplay=swap"

rel="stylesheet"

/>

<link

href="https://fonts.googleapis.com/css2?family=Crimson+Text&display=swap" rel="stylesheet"

/>

</head>

<body>

<div class="form">

<form>

<br /><br />

<div class="btncont">

<button style="margin-right: -2%" type="button">

NEWSTRACKER APP

</button>

<button

style="margin-left: 40%" type="button"

onclick="window.location.href='/logout'"

>

LOGOUT

</button>

<button type="button" onclick="window.location.href='/dashboard'">

DASHBOARD

</button>

</div>

</form>

</div>

<br /><br /><br /><br />

{% for news, author,publishedAt, desc,img,content,url in context %}

<br />

<div class="cont">

<center><h1>{{news}}</h1></center>

<img style="height: 464px; width: 100%" src="{{img}}" alt="" /><br />

<h2>Published Time: {{publishedAt}}</h2>

<h2>Author: {{author}}</h2>

<br />

<h3>Description: {{desc}}</h3>

<br />

<h3>

Content: {{content}}<a href="{{url}}" target="blank">Read More...</a>

</h3>

<br />

</div>

{% endfor %}

</body>

</html>

SIGNUP:

<html>

<head>

<title>SIGN UP</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta charset="utf-8" />

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='pages.css') }}" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css"/>

<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottieplayer.js"></script>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-

vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp" crossorigin="anonymous">

</head>

<body class="body">

<div class="on-screen">

<div class="form">

<form action="/register" method="POST">

<input type="text" name="fullname" placeholder="Full Name" required/>

<input type="email" name="email" placeholder="Email address" required/>

<input type="text" name="username" placeholder="Username" required/>

<input type="password" name="password" id="password" placeholder="Set a password" required/><br>

<i class="fas fa-eye" onclick="show()"></i><br>

<h4>{{msg}}</h4><br>

<button style="width:135px" type="submit">SUBMIT</button>

<button style="width:135px" type="button" onclick="window.location.href='/'">LOGIN</button>

</form>

</div>

</div>

</body>

<script>

*function* show()

{

*var* password = document.getElementById("password"); *var* icon = document.querySelector(".fas"); if (password.type === "password")

{

password.type = "text";

} else

{

password.type = "password";

}

}

</script>

</html>

DASHBOARD:

<html>

<head>

<title>DASHBOARD</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='dashboard.css')

}}">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">

<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottieplayer.js"></script>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.1/css/all.css" integrity="sha384-

vp86vTRFVJgpjF9jiIGPEEqYqlDwgyBgEF109VFjmqGmIY/Y4HV4d3Gp2irVfcrp" crossorigin="anonymous">

</head>

<body class="body">

<div class="on-screen">

<div class="form">

<form action="/news" class="innercont" method="POST">

<h4>Welcome,{{username}}</h4>

<center><h4>Select one of the below sources.</h4></center>

<button type="submit"name="newsresource" value="google"><img

name="newsresource" value="abc"src="https://ibm-news-tracker.s3.jp-tok.cloud-objectstorage.appdomain.cloud/google.jpg" /></button>

<button type="submit"name="newsresource" value="bbc"><img

name="newsresource" value="abc"src="https://ibm-news-tracker.s3.jp-tok.cloud-objectstorage.appdomain.cloud/bbc.jpg" /></button><br><br>

<button type="submit"name="newsresource" value="toi"><img name="newsresource" value="abc"src="https://ibm-news-tracker.s3.jp-tok.cloud-objectstorage.appdomain.cloud/toi.jpg" /></button>

<button type="submit"name="newsresource" value="abc"><img

name="newsresource" value="abc"src="https://ibm-news-tracker.s3.jp-tok.cloud-objectstorage.appdomain.cloud/abc.jpg" /></button>

<p class="message"></p>

</form>

</div>

</div>

</body>

</html>

FLASK APP:

from flask import Flask,render\_template,request,redirect,url\_for,session from flask\_mail import Mail, Message from newsapi import NewsApiClient import ibm\_db import re app=Flask(\_\_name\_\_) mail = Mail(app)

app.config['MAIL\_SERVER']='smtp.gmail.com' app.config['MAIL\_PORT'] = 465

app.config['MAIL\_USERNAME'] = 'news.tracker.app.1@gmail.com' app.config['MAIL\_PASSWORD'] = 'Oms@1000' app.config['MAIL\_USE\_TLS'] = False app.config['MAIL\_USE\_SSL'] = True mail = Mail(app) app.secret\_key = 'a'

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=

SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fvs98779;PWD=d2GpGOw7rU073ye

E",' ',' ')

global newsresource

@app.route('/') *def* home():

return render\_template('login.html')

@app.route('/signup') *def* signup():

return render\_template('signup.html')

@app.route('/forgot') *def* forgot():

return render\_template('forgot.html')

@app.route('/dashboard') *def* dashboard(): if(session['loggedin']==True): return render\_template('dashboard.html',*username*=session['fullname']) return render\_template('login.html')

@app.route('/login',*methods*=['GET','POST']) *def* login(): global userid msg=' ' if request.method=='POST': username = request.form['username'] password = request.form['password']

sql = "SELECT \* FROM NTAAC WHERE username = ? AND password = ?" stmt = ibm\_db.prepare(conn,sql) ibm\_db.bind\_param(stmt,1,username) ibm\_db.bind\_param(stmt,2,password) ibm\_db.execute(stmt)

account = ibm\_db.fetch\_assoc(stmt) if account: session['loggedin'] = True session['id'] = account['USERNAME'] userid= account['USERNAME']

session['username'] = account['USERNAME'] session['fullname']=account['FULLNAME'] session['email'] = account['EMAIL']

msg = 'Logged in successfully !'

return render\_template('dashboard.html',*username*=session['fullname']) else:

msg = 'Incorrect username / password !' return render\_template('login.html',*msg*=msg)

@app.route('/register',*methods*=['GET','POST']) *def* register(): if request.method=='POST': fullname = request.form['fullname'] username = request.form['username'] email = request.form['email'] password = request.form['password']

queryone = "SELECT \* FROM NTAAC WHERE username = ? " stmtone = ibm\_db.prepare(conn,queryone) ibm\_db.bind\_param(stmtone,1,username) ibm\_db.execute(stmtone)

userexist = ibm\_db.fetch\_assoc(stmtone) querytwo = "SELECT \* FROM NTAAC WHERE email = ? " stmtwo = ibm\_db.prepare(conn,querytwo) ibm\_db.bind\_param(stmtwo,1,email) ibm\_db.execute(stmtwo)

emailexist = ibm\_db.fetch\_assoc(stmtwo) if emailexist:

msg="Email id already exist!" elif userexist:

msg="Username id already exist!" else:

insert\_sql="INSERT INTO NTAAC VALUES(?, ?, ?, ?)" prep\_stmt=ibm\_db.prepare(conn,insert\_sql) ibm\_db.bind\_param(prep\_stmt,1,fullname) ibm\_db.bind\_param(prep\_stmt,2,email) ibm\_db.bind\_param(prep\_stmt,3,username) ibm\_db.bind\_param(prep\_stmt,4,password) ibm\_db.execute(prep\_stmt)

# msg = Message('CAD-NEWSTRACKER Account Created Successfully',sender

='cad.newstracker@gmail.com',recipients = [email])

# msg.body = ("Welcome to CAD-NEWSTRACKER, your account was successfully registered with us."+"\n\n\n"+"Your login credentials are:"+"\n\n"+"Username:

"+username+"\n"+"Password: "+password+"\n\n\n"+"Happy

Reading........"+"\n\n\n\n"+"With regard,"+"\n"+"CAD-NEWSTRACKER")

# mail.send(msg)

return render\_template('newaccount.html') return render\_template('signup.html',*msg*=msg)

@app.route('/recover',*methods*=['GET','POST']) *def* recover(): if request.method=='POST':

email = request.form['email']

query = "SELECT \* FROM NTAAC WHERE email = ? "

stmt = ibm\_db.prepare(conn,query) ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

emailexist = ibm\_db.fetch\_assoc(stmt) if emailexist:

queryone=("SELECT \* FROM NTAAC WHERE EMAIL =?") stmtone = ibm\_db.prepare(conn,queryone) ibm\_db.bind\_param(stmtone,1,email)

ibm\_db.execute(stmtone)

credentials = ibm\_db.fetch\_assoc(stmtone) username=str(credentials['USERNAME']) password=str(credentials['PASSWORD'])

# msg = Message('CAD-NEWSTRACKER Login Credentials',sender

='charanya@gmail.com',recipients = [email])

# msg.body = ("Request for sending your login credentials was completed successfully."+"\n\n\n"+"Your login credentials are:"+"\n\n"+"Username: "+username+"\n"+"Password: "+password+"\n\n\n"+"Happy

Reading........"+"\n\n\n\n"+"With regard,"+"\n"+"CAD-NEWSTRACKER")

# mail.send(msg)

msg="Registered succcessfully....." email=' '

return render\_template('login.html',*msg*=msg) else:

msg="Email id not found."

return render\_template('forgot.html',*msg*=msg)

@app.route('/news',*methods*=['GET','POST']) *def* news():

newsapi = NewsApiClient(*api\_key*="15fc18be3b65448987ae91aa3bcb50d4") if request.method == 'POST': if(request.form['newsresource']=="google"): newsresource="google-news-in" msg="GOOGLE NEWS" elif(request.form['newsresource']=="bbc"):

newsresource="bbc-news" msg="BBC NEWS" elif(request.form['newsresource']=="toi"): newsresource="the-times-of-india" msg="Times of India" elif(request.form['newsresource']=="abc"): newsresource="abc-news" msg="ABC NEWS"

topheadlines = newsapi.get\_top\_headlines(*sources*=newsresource) articles = topheadlines['articles'] news = [] author = [] publishedAt=[] desc=[] img = [] content=[] url=[] for i in range(len(articles)): myarticles = articles[i] news.append(myarticles['title']) author.append(myarticles['author']) publishedAt.append(myarticles['publishedAt']) desc.append(myarticles['description']) img.append(myarticles['urlToImage']) content.append(myarticles['content']) url.append(myarticles['url'])

mylist = zip(news, author,publishedAt, desc,img,content,url) return render\_template('news.html', *context* = mylist)

@app.route('/logout') *def* logout():

session['loggedin']=False session.pop('id', None) session.pop('username', None) msg="Logged out successfully......" return render\_template('login.html',*msg*=msg)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(*host*='0.0.0.0', *port*=5000, *debug*=True, *threaded*=True)

**Result:**

News tracker application using cloud is developed and executed at the level of completed progress .

#### 10. ADVANTAGES & DISADVANTAGES Advantages:-

* In this app news is already categorization.
* Easily accessible and portable.
* Better user experience.
* Apps help you convert visitors to loyal readers.
* Minute by minute updates of news.
* This app helps you to get local news updates instantly.
* To explore and discover trending news and topics. ❖ News feeds for you based on your interest.

## Disadvantages:-

* Some apps demand premium subscription from user.
* Occurance of Advertisment disturb the user. ❖ Sometimes the news gives brief information.

❖

Prevailance of fake and uncertain news can confuse the user lead to misconception.

❖

Fake news may mislead the readers.

### 11. CONCLUSION

We explored the feasibility of recognising patterns of news reading interactions and evaluated three adaptive interface designs for different news reader types. We show that from their interaction log, a specific user can be recognised as one of three kinds. The reader types emerging from the online survey are well defined and distinct. The evaluation of the three variant interfaces suggests that different news reader types need different user interfaces. We have demonstrated a method for monitoring users’ news reading behaviour and inferring news reader type from it. In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

## 12.FUTURE SCOPE

In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

**GitHub**

## <https://github.com/IBM-EPBL/IBM-Project-13148-1659511897>

## 