PROJECT REPORT

NEWS TRACKER APPLICATION

submitted by

PNT2022TMID27314

PRAVEEN KUMAR R - 311019104063
SUPRIYA M - 311019104092
VISHWA T- 311019104100
YUDHASINI S- 311019104103

TABLE OF CONTENTS

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX

Source Code

GitHub & Project Demo Link

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

This app is a news application. This application will show the news about the world. This is better than old conventional newspapers as you can see news any time and anywhere now. This will also have live updates.

Users can see the real news with real-time updates.The app will show category-wise news.Auto update news because it fetch news from API.Users can see the news according to their interests.

1.2 PURPOSE

Today, the publishing industry is facing such a threat when it comes to newspaper publishing and sales. So, magazine and newspaper lovers are moving towards reading news on mobiles and tablets. The revenue model of the online apps are quite simple and rewarding.So.users can view the updated news on time in their own hands.They can view the news depend upon their own interest.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fundamental problem with news apllications is there are multiple news-sharing apps used by a single user and are often spammed with notifications. There is also a lot of fake news which gets shared. A news-sharing app wants to help users find relevant and important news easily everyday and also understand explicitly that the news is not fake but from proper sources.Users are facing a lot of advertisement in between the news reading and more unwanted news which beyond their interests are shown in the feeds.

2.2 REFERENCES

[1] https://www.ripublication.com/ijaer18/ijaerv13n11_78.pdf

[2] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 9310-9315 Research India Publications. http://www.ripublication.com

[3] Sangeeta Ruth, Srividhya Raghavan V, Smrithi J,Saira Banu. 2016. “Spatial Preference NewsfeedSystem For Android Mobile Users”, IJCSITS, Vol6, NO. 3: 24.

[4] https://newsapi.org/

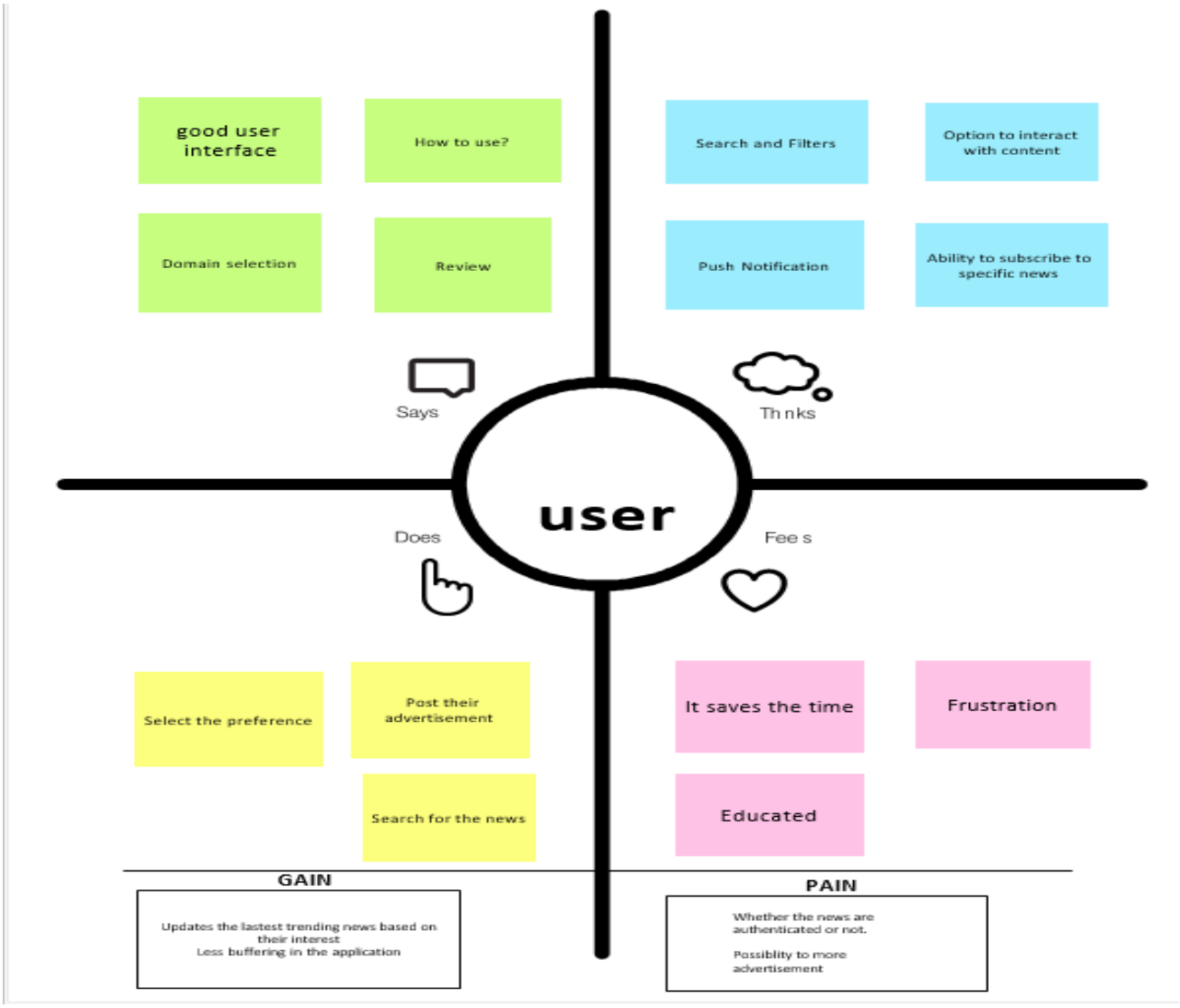
[5] https://dzone.com/articles/how-to-parse-json-datafrom-a-rest-api-using-simpl

2.3 PROBLEM STATEMENT DEFINITION

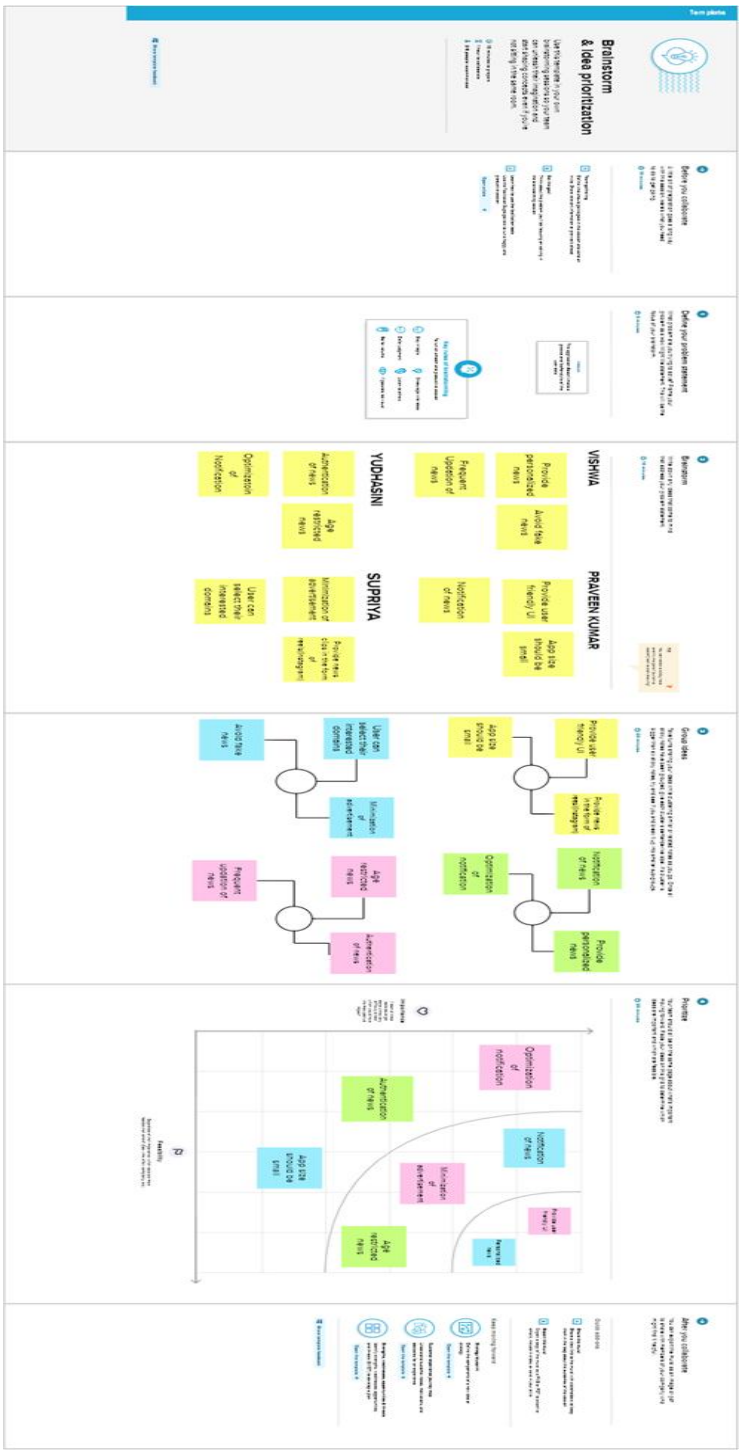
The study presents a personalized e-news monitoring agent system which employs the topic tracking based approach for tracking the user-interested news events. The proposed scheme simultaneously considers both the similarities and the semantic relationships among news topics to track the user-interested news topics.Storing user details and their preferences in the cloud platform. Cloud computing is one of the upcoming technologies that will upgrade generation of Internet. The data stored in the smart phones is increased as more applications are deployed and executed. If the phone is damaged or lost then the information stored in it gets lost. If the cloud storage can be integrated for regular data backup of a mobile user so that the risk of data lost can be minimized.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To provide customized news with current updation.
2.	Idea / Solution description	Provide user friendly UI Avoid fake News User privacy and security Improve app speed Create own wishlist
3.	Novelty / Uniqueness	This app have separate news domain and have minimal advertisement. Provide news clips in the form of short videos.
4.	Social Impact / Customer Satisfaction	The user able to get news with minimal ads. The user will not have to spend time on searching news rather than get updated news from their Wishlist.
5.	Business Model (Revenue Model)	The user can switch to premium to get news without ads. Organisations can approach and post their advertisement in our application.
6.	Scalability of the Solution	Since the web application is deployed on IBM cloud , it can handle multiple user at a time. The user view the news according to their interest and choice. User from all age category can use the application and the news can also be filtered according to their age.

3.4 PROBLEM SOLUTION FIT

Project Title:NEWS TRACKER APPLICATION **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID27314

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Users should login the app with the User's name or email and password.
FR-4	User Information	Users can include their information in preferred topics so that they could be recommended by the application.

4.2 NON FUNCTIONAL REQUIREMENTS

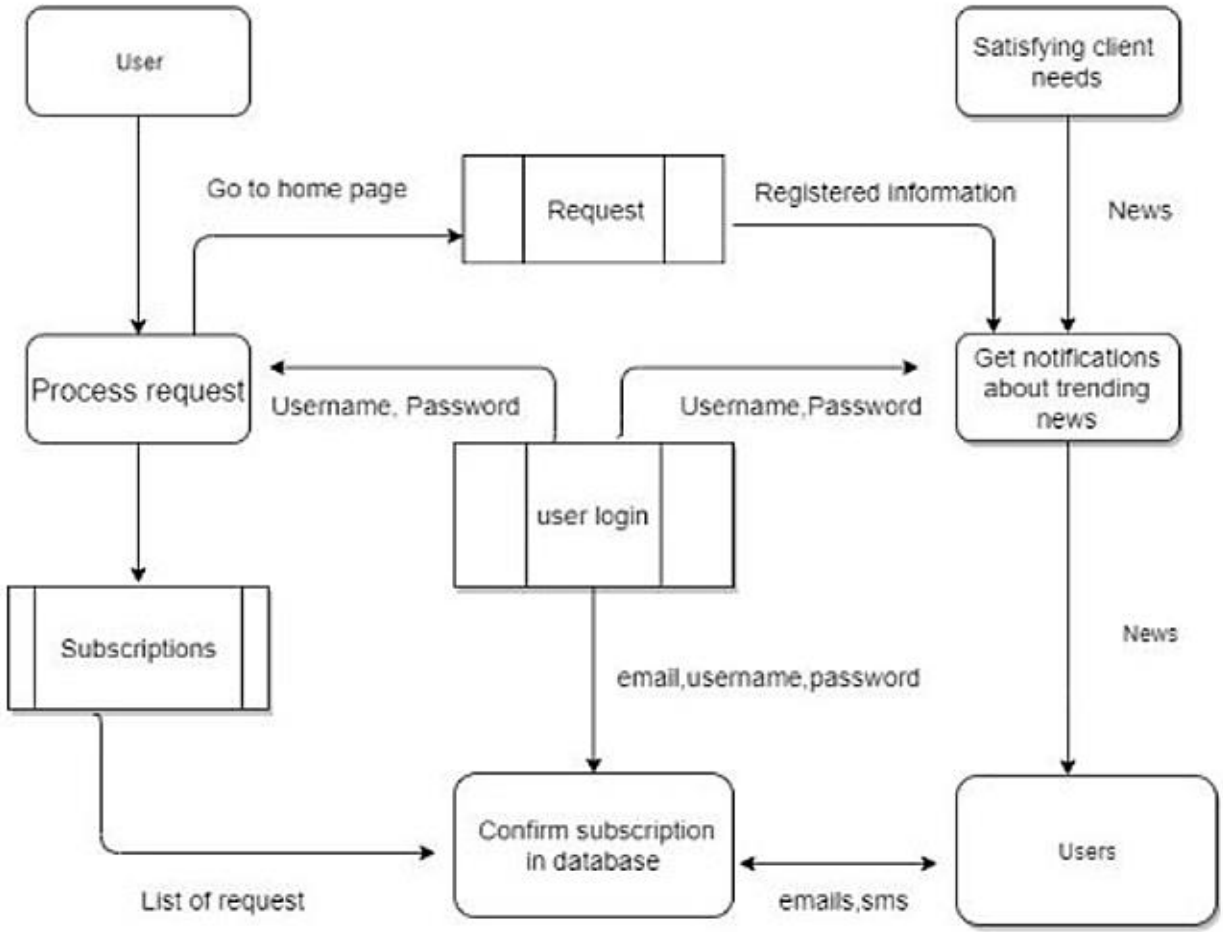
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This app is user friendly app, where users can clear their queries with the chatbot.
NFR-2	**Security**	This app is secured app, where users information is encrypted properly.
NFR-3	**Reliability**	This app can be accessed anywhere and anytime. User can download the news offline.
NFR-4	**Performance**	The app is well tested and hence the performance of the app is great.
NFR-5	**Availability**	Chatbot is available in this app to rectify the queries of the users
NFR-6	**Scalability**	This app refers to the capacity of an app to handle growth.

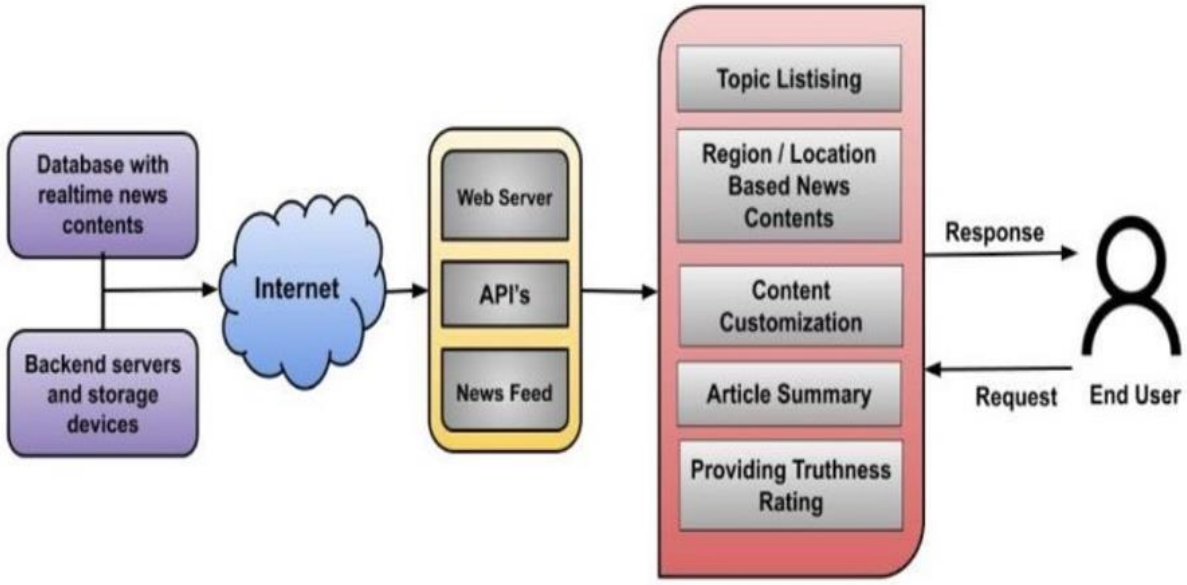
CHAPTER 5
PROJECT DESIGN

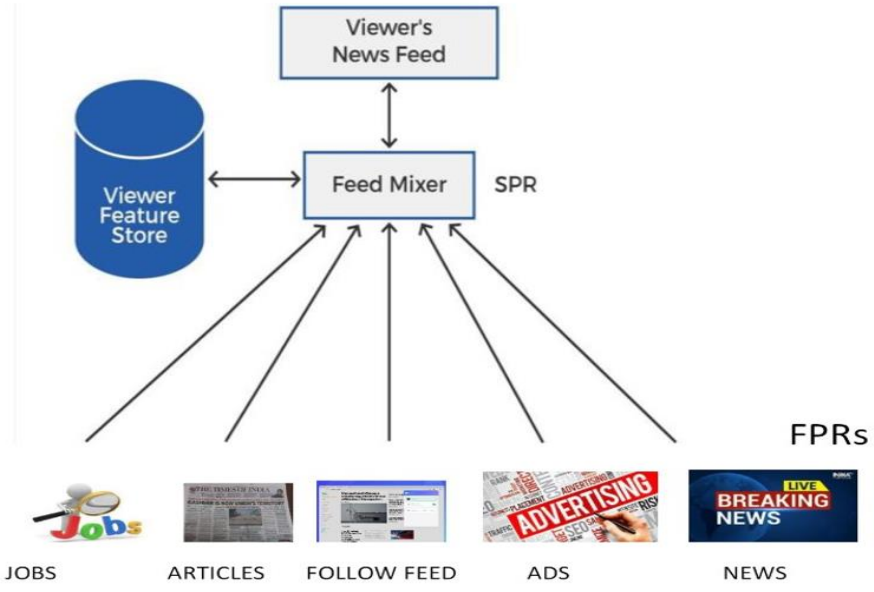
5.1 DATA FLOW DIAGRAM

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 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

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	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
		USN-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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive password from mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can view all types of information through this application	High	Sprint-1
	Dashboard	USN-6	The news portal fetches for the most recent news and shows it as "Breaking News"	I can open and view the "Breaking News"	Low	Sprint -2
Customer (Web user)	Browser	USN-7	Have interactive medium between client and server	I have a clarity to use this application and easily resolve my specific issues	High	Sprint -1
Customer Care Executive	Chat box	USN-8	Rectify the issues related subscription, account, terms and conditions, privacy policy			
Administrator	Registration with account	USN-9	As a user, I can simply register with my Facebook account	I can access my account / Profile.	Medium	Sprint -1
	Registration with domain	USN-10	As a user, I can register to the website with my Gmail account	I can access my account / Profile.	Medium	Sprint -1
	Login	USN-11	As a user, I can register to the website with my registered account	I can access my account / Profile.	Medium	Sprint -1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND 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	Vishwa T
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Yudhasini S
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	Supriya M
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Praveen Kumar R
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Vishwa T
Sprint-2	Dashboard	USN-6	As a user I can select the news domains based on the wish.	1	High	Supriya M
Sprint-2		USN-7	As a user I can see the news in the dashboard.	2	High	Yudhasini S
Sprint-3	Search bar	USN-8	As a user I can search the news.	2	Medium	Praveen Kumar R
Sprint-3		USN-9	As a user I can get notified about the news.	2	High	Vishwa T
Sprint-4	Admin login	USN-10	As a admin I can manage news and user logins	1	Medium	Vishwa T
Sprint-4		USN-11	As a admin I can check authenticity of news	1	Medium	Yudhashini S
Sprint-4		USN-12	As a user I can switch to premium account.	3	Medium	Praveen Kumar R

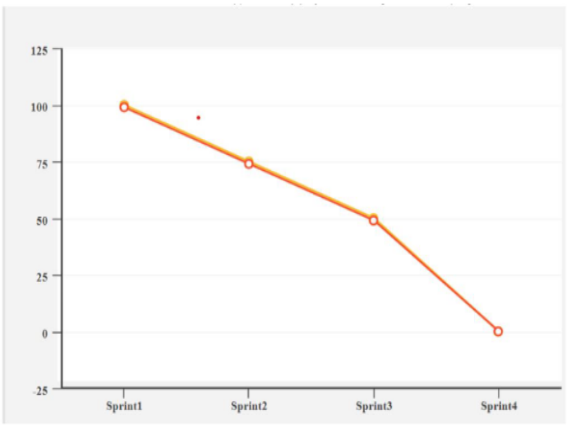
6.2 SPRINT DELIVERY SCHEDULE

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	10	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	10	14 Nov 2022

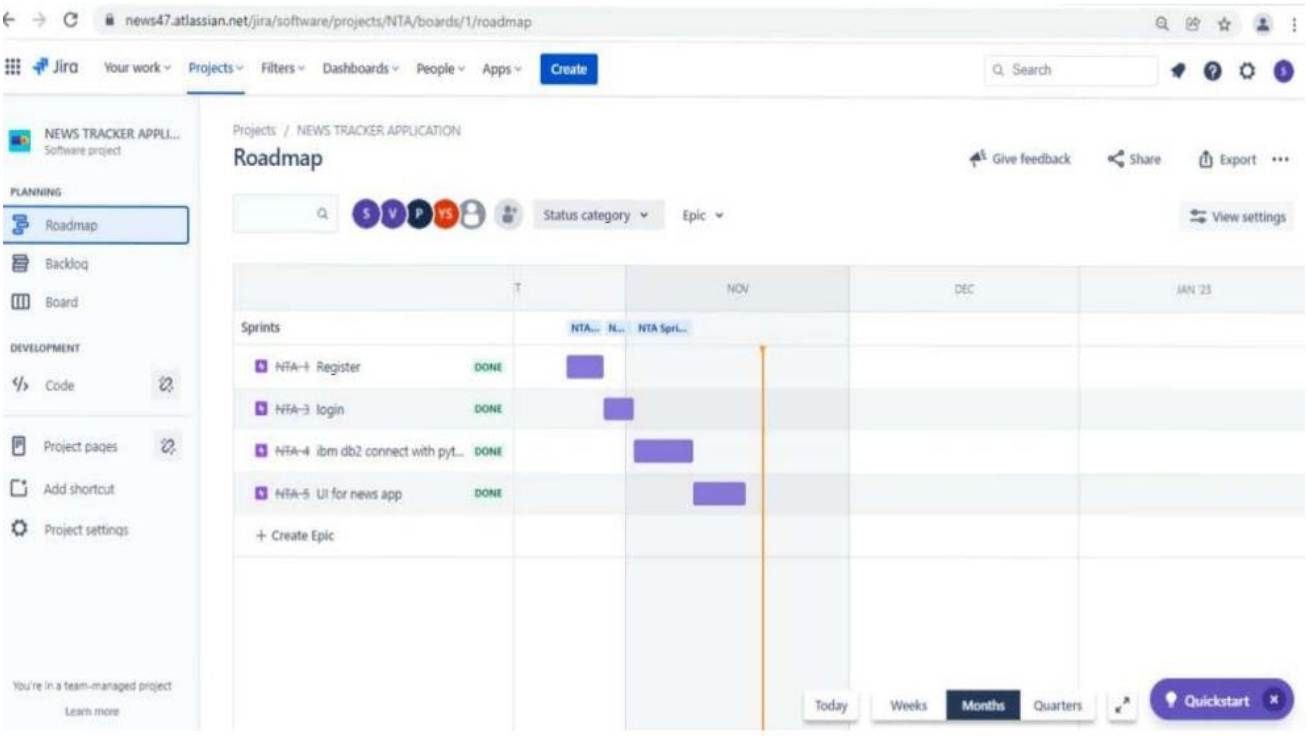
Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 Reports from JIRA



CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re

app = Flask(__name__)

hostname = 'ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'lhm02447'
pwd = 'GuPGCnMIAXaExvbo'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'bludb'
port = '31321'
protocol = 'TCPIP'
cert = "C:/Users/Prithiarun/Desktop/IBM/TEST/certi.crt"
dsn = (
    "DATABASE ={0};"
    "HOSTNAME ={1};"
    "PORT ={2};"
    "UID ={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "PWD ={6};"
).format(db_name, hostname, port, uid, protocol, cert, pwd)
connection = ibm_db.connect(dsn, "", "")
print()
# query = "SELECT username FROM USER1 WHERE username=?"
# stmt = ibm_db.prepare(connection, query)
# ibm_db.bind_param(stmt, 1, username)
# ibm_db.execute(stmt)
# username = ibm_db.fetch_assoc(stmt)
# print(username)
app.secret_key = 'a'



@app.route('/register', methods=['GET', 'POST'])
def register():
    return render_template('register.html')
```

```
@app.route('/insert', methods=['GET', 'POST'])
def insert():
    msg = " "
    if request.method == 'POST':
        username = request.form['name']
        email_id = request.form['email']
        password = request.form['password']
        query = "SELECT * FROM USER WHERE name=?;"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if (account):

            msg = "Account already exists!"
            return render_template('register.html', msg=msg)
        # elif not re.match(r'[^@]+@[^@]+\.[^@]+', email_id):
        #     msg = "Invalid email addres"
        # elif not re.match(r'[A-Za-z0-9+', username):
        #     msg = "Name must contain only characters and numbers"
        else:
            query = "INSERT INTO USER values(?,?,?)"
            stmt = ibm_db.prepare(connection, query)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email_id)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.execute(stmt)
            msg = 'You have successfully Logged In!!'
            return render_template('login.html', msg=msg)
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
        return render_template('register.html', msg=msg)

@app.route('/', methods=['GET', 'POST'])
@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ' '
```

```
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password']
        query = "select * from user where mail=? and password=?"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin'] = True
            session['id'] = account['NAME']
            session['username'] = account['NAME']
            msg = 'Logged in Successfully'
            return render_template('index.html', msg=msg, username=str.upper( account['NAME']))
        else:
            msg = 'Incorrect Username or Password'
            return render_template('login.html', msg=msg)
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
        return render_template('login.html', msg=msg)


@app.route('/welcome', methods=['GET', 'POST'])
def welcome():
    if request.method == 'POST':
        username = request.form['username']
        print(username)
        return render_template('index.html', username=username)
    else:
        return render_template('index.html', username=username)


if __name__ == "__main__":
    app.run(debug=True)
    app.run(host='0.0.0.0')
```

7.2 Feature 2

```
body {
    background-color: hsl(39, 77%, 83%,0.2);
    font-family: "Roboto", sans-serif;
  }

  .signup-box {
    width: 360px;
    height: 620px;
    margin: auto;
    background-color: hsl(39, 77%, 83%,0.4);
    border-radius: 3px;
  }

  .login-box {
    width: 360px;
    height: 280px;
    margin: auto;
    border-radius: 3px;
    background-color: hsl(39, 77%, 83%,0.4);
  }

  h1 {
    text-align: center;
    padding-top: 15px;
  }

  h4 {
    text-align: center;
  }

  form {
    width: 300px;
    margin-left: 20px;
  }

  form label {
    display: flex;
    margin-top: 20px;
    font-size: 18px;
```

```
}

form input {
  width: 100%;
  padding: 7px;
  border: none;
  border: 1px solid gray;
  border-radius: 6px;
  outline: none;
}
input[type="button"] {
  width: 320px;
  height: 35px;
  margin-top: 20px;
  border: none;
  background-color: #49c1a2;
  color: white;
  font-size: 18px;
}
p {
  text-align: center;
  padding-top: 20px;
  font-size: 15px;
}
.para-2 {
  text-align: center;
  color: black;
  font-size: 15px;
  margin-top: -10px;
}
.para-2 a {
  color: #49c1a2;
}

.box{

  margin-left: 40%;
}
```

7.3 Database Schema

```
hostname = 'ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'lhm02447'
pwd = 'GuPGCnMIAXaExvbo'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'bludb'
port = '31321'
protocol = 'TCPIP'
cert = "C:/Users/Prithiarun/Desktop/IBM/TEST/certi.crt"
dsn = (
    "DATABASE ={0};"
    "HOSTNAME ={1};"
    "PORT ={2};"
    "UID ={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "PWD ={6};"
).format(db_name, hostname, port, uid, protocol, cert, pwd)
connection = ibm_db.connect(dsn, "", "")
print()
# query = "SELECT username FROM USER1 WHERE username=?"
# stmt = ibm_db.prepare(connection, query)
# ibm_db.bind_param(stmt, 1, username)
# ibm_db.execute(stmt)
# username = ibm_db.fetch_assoc(stmt)
# print(username)


 query = "SELECT * FROM USER WHERE name=?;"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

LOGIN

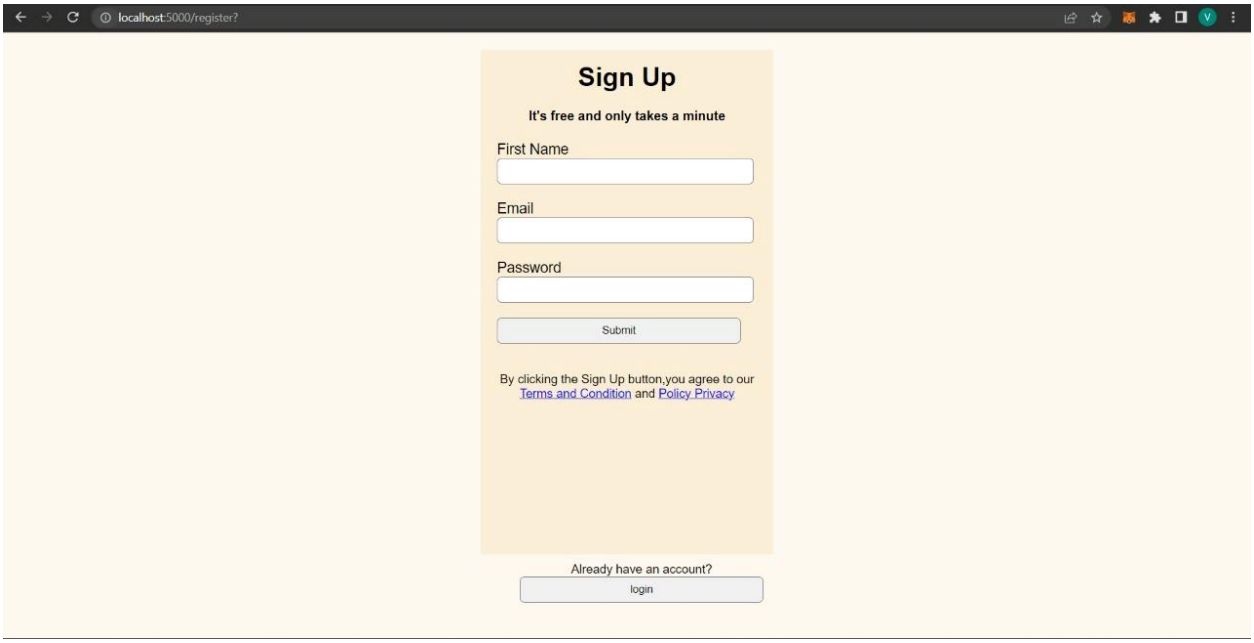
```
<!DOCTYPE html>
<html lang="en">
 <head>
   <title>Login | By Code Info</title>
   <!--<link rel="stylesheet" href="/Static/css/style.css"    />-->
   <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}" />
   <link
    href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap"
    rel="stylesheet"
   />
 </head>
 <body>
   <div class="login-box">
    <h1>Login</h1>
    <form action="/login" method="POST">
     <label>Email</label>
     <input type="email" name="email" placeholder="" />
     <label>Password</label>
     <input type="password"  name="password" placeholder="" />
     <input type="submit" value="Submit" />
    <closeform></closeform></form>
   </div>

  <div class="box" >

   <form action="/register">

    <p class="para-2">
     Not have an account?   <input   value="sign-up" type="submit">
    </p>


   </form>

      {{msg}}
  </div>
  </body>
</html>
```

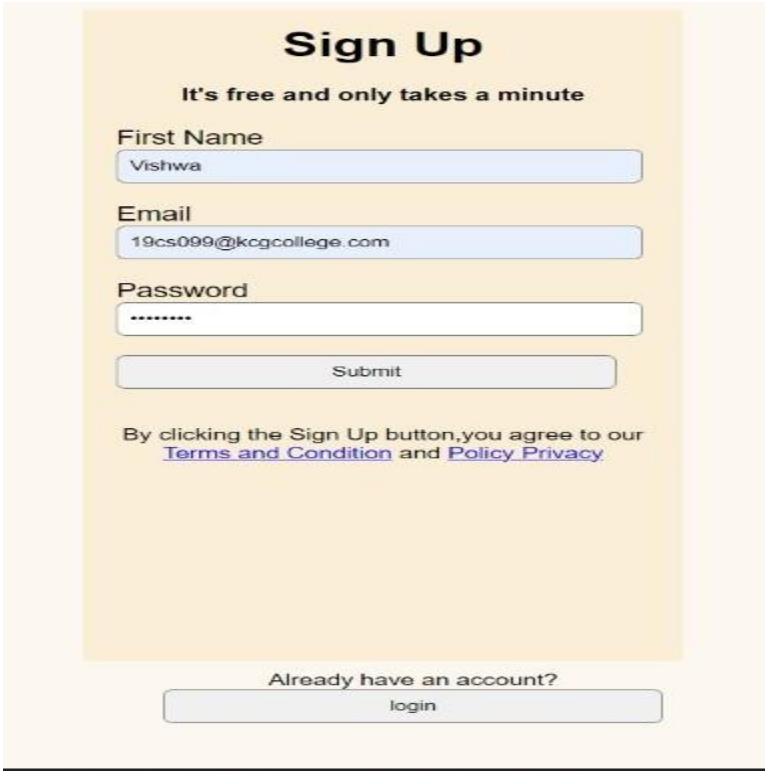

Login
Email
Password
Submit
Not have an account?
sign-up
PLEASE FILL OUT OF THE FORM

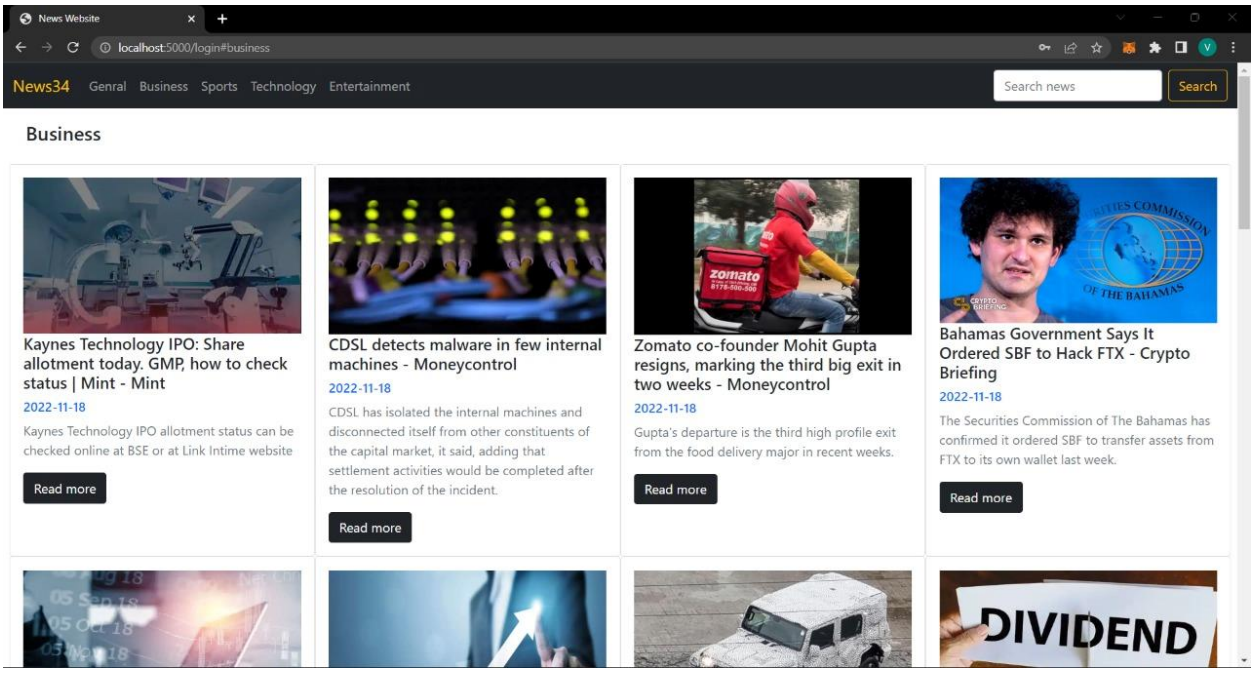

Login
Email
19cs099@kcgcollege.com
Password
Submit
Not have an account?
sign-up
PLEASE FILL OUT OF THE FORM

REGISTER

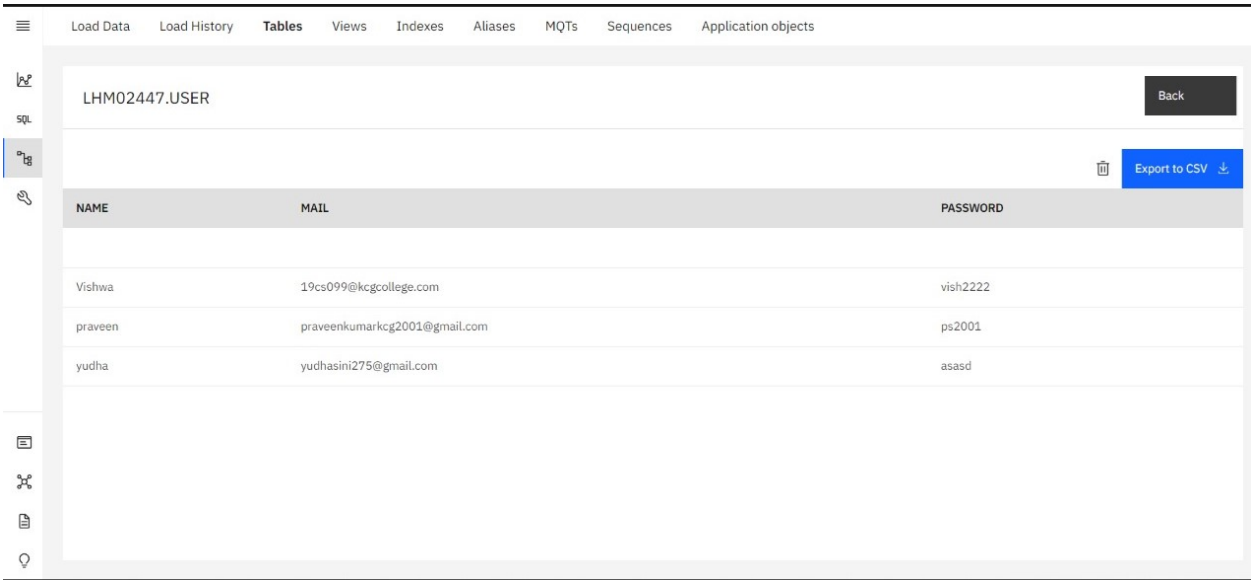
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Sign Up | By Code Info</title>
   <!--<link rel="stylesheet" href="/Static/style.css" />-->
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}" />
  <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap"
      rel="stylesheet"/>
  </head>
  <body>
    <div class="signup-box">
      <h1>Sign Up</h1>
      <h4>It's free and only takes a minute</h4>
      <form action="/insert" method="post">
        <label>First Name</label>
        <input  name="name" type="text" placeholder="" />
        <label>Email</label>
        <input  name="email" type="email" placeholder="" />
        <label>Password</label>
        <input name="password" type="password" placeholder="" />
       <br>
       <br>
      <input   type="submit" value="Submit" />
      <closeform></closeform></form>
      <p>
        By clicking the Sign Up button,you agree to our <br />
        <a href="#">Terms and Condition</a> and <a href="#">Policy Privacy</a>
      </p>
    </div>
    <div  class="box">

      <form     action="/login">
        <p class="para-2">
          Already have an account? <input  value="login" type="submit">
        </p>
   </form>
 </div>
   </body>
</html>
```


localhost:5000/register?
Sign Up
It's free and only takes a minute
First Name
Email
Password
Submit
By clicking the Sign Up button,you agree to our
Terms and Condition and Policy Privacy
Already have an account?
login


Sign Up
It's free and only takes a minute
First Name
Vishwa
Email
19cs099@kcgcollege.com
Password
Submit
By clicking the Sign Up button,you agree to our
Terms and Condition and Policy Privacy
Already have an account?
login



Cloud Storage

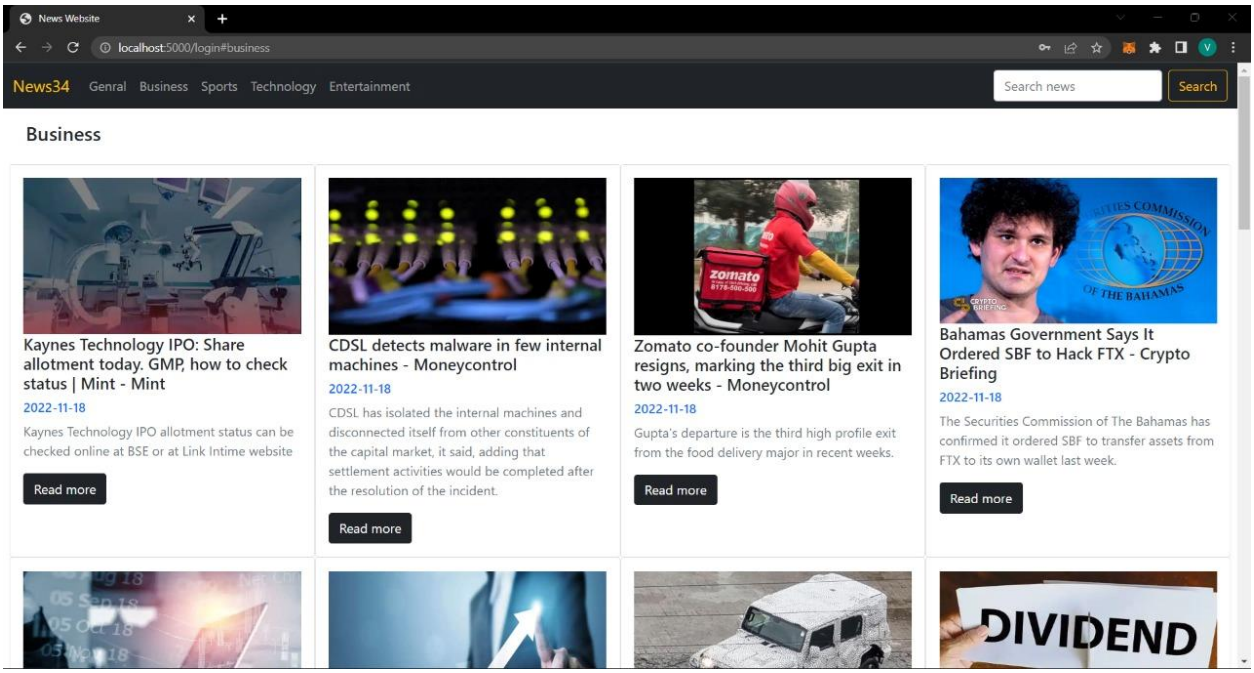


CHAPTER 8

TESTING

8.1 Test Cases

The user can enter the application and select the domains he want and he can also search the news he wants to know.



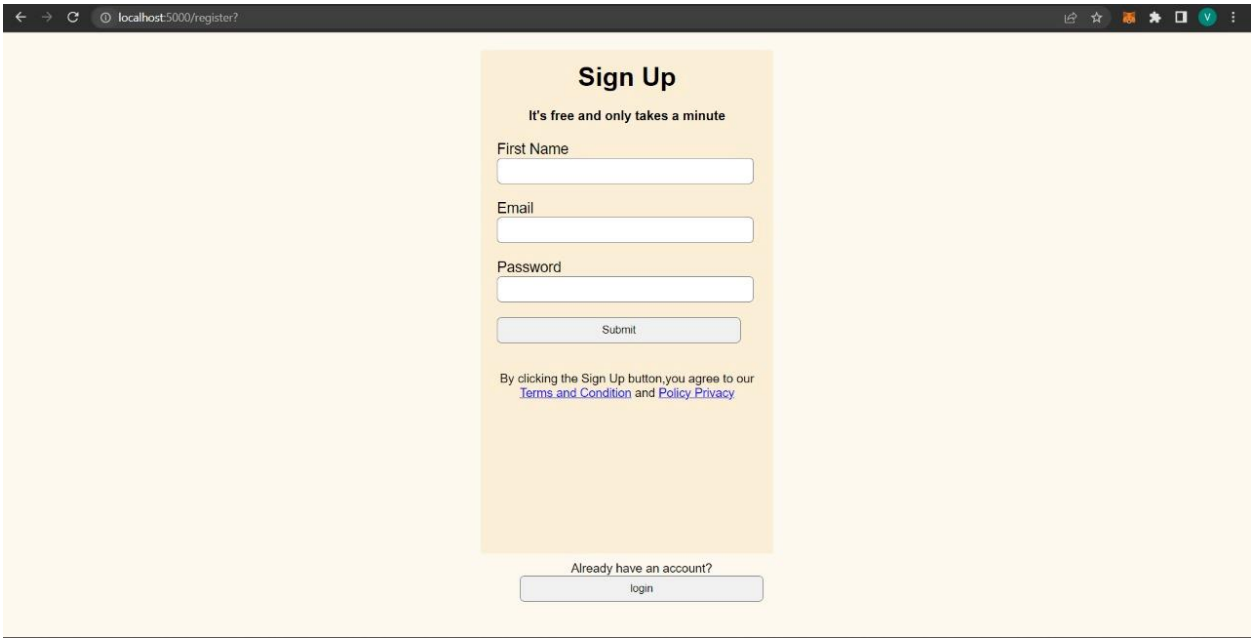
8.2 User Acceptance Testing

CHAPTER 9

RESULTS

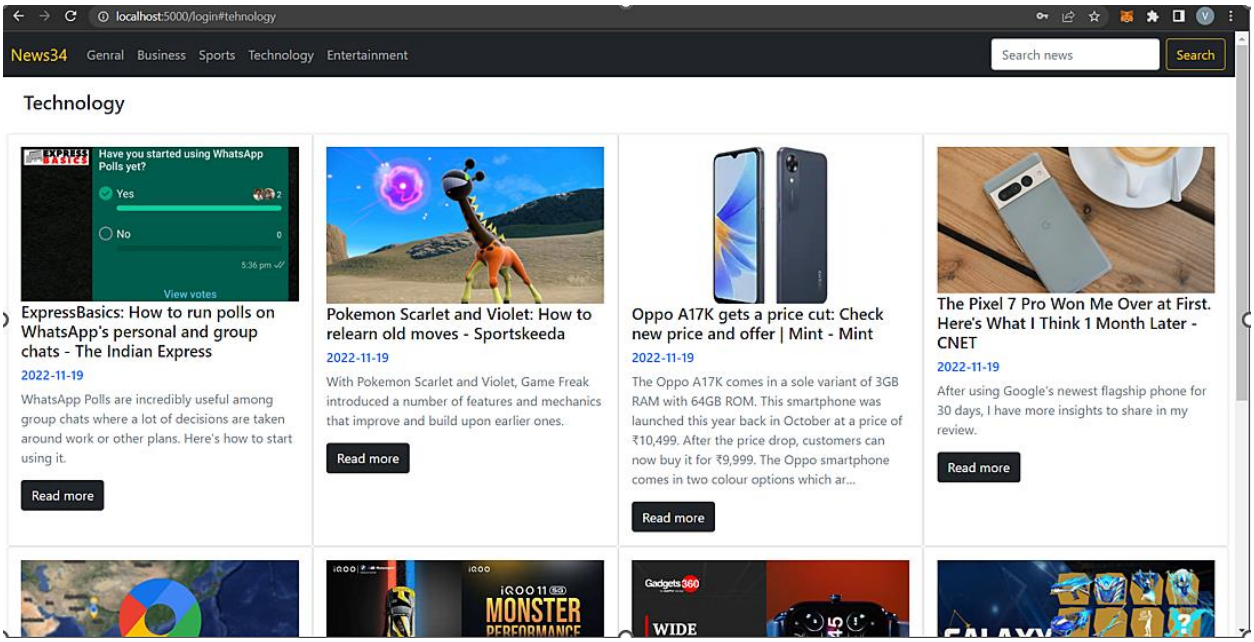
9.1 Performance Metrics

REGISTER:

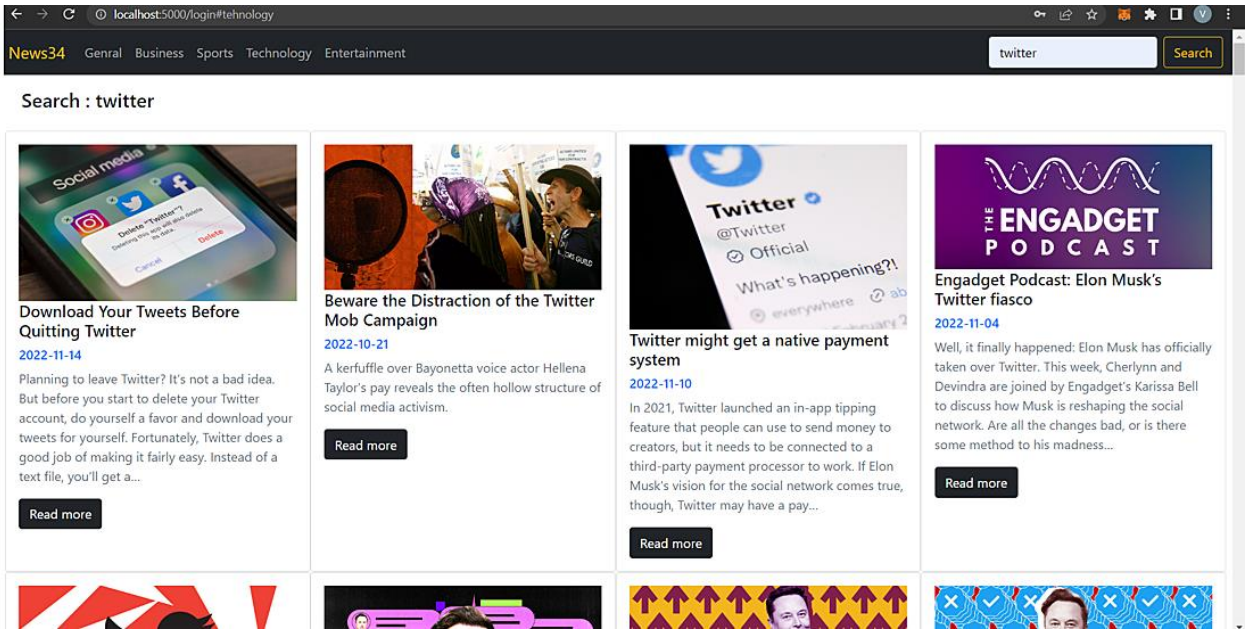


LOGIN

DASHBOARD



SEARCH NEWS:


localhost:5000/login#tehnology
News34
Genral
Business
Sports
Technology
Entertainment
twitter
Search
Search : twitter
Download Your Tweets Before Quitting Twitter
2022-11-14
Planning to leave Twitter? It's not a bad idea. But before you start to delete your Twitter account, do yourself a favor and download your tweets for yourself. Fortunately, Twitter does a good job of making it fairly easy. Instead of a text file, you'll get a...
Read more
Beware the Distraction of the Twitter Mob Campaign
2022-10-21
A kerfuffle over Bayonetta voice actor Hellena Taylor's pay reveals the often hollow structure of social media activism.
Read more
Twitter might get a native payment system
2022-11-10
In 2021, Twitter launched an in-app tipping feature that people can use to send money to creators, but it needs to be connected to a third-party payment processor to work. If Elon Musk's vision for the social network comes true, though, Twitter may have a pay...
Read more
Engadget Podcast: Elon Musk's Twitter fiasco
2022-11-04
Well, it finally happened: Elon Musk has officially taken over Twitter. This week, Cherlynn and Devindra are joined by Engadget's Karissa Bell to discuss how Musk is reshaping the social network. Are all the changes bad, or is there some method to his madness...
Read more

CHAPTER 10
ADVANTAGES & DISADVANTAGES

Advantages:

1. The user will be updated

The application fetches the news from the API which Rapidly updates the news.

2.Saves time

The news were shown in a card like API so that the user skim the news very easily.

3.Domain switch

The user can switch between the domains very easily as Ui is made for that process.

4.Minimal Add

The add disturbance will be minimal.By swithcing to premium account we can avoid the adds totally.

Disadvantages:

1. There may be server problem due to over load.
2. There is add eventhough it is minimum
3. Irrelevant news can be shown sometimes.

CHAPTER 11
CONCLUSION

This application is entirely user control.And this application helps the user to keep them updated. Many of the features are customized. User can change their news domains as per their interest.

CHAPTER 12
FUTURE SCOPE

1. It will have Notification system.
2. It will have premium account switching option.
3. There will be a admin to check the authenticity of news.
4. News will be displayed based on age restriction.

CHAPTER 13
APPENDIX

Source Code

GitHub ***:https://github.com/IBM-EPBL/IBM-Project-12780-1659464121.git***

Project Demo Link:
https://drive.google.com/file/d/1cMDPf_eoo2s7nRj6_R2SfILcbtSbG6R2/view?usp=share_link