

Project Name

NewsTrackeApplication

Domain

CloudApplicationDevelopment

Team ID

PNT2022TMID49391

Team Leader

SIVAGAMASUNDARI S

Team Member 1

DEVIKA L

Team Member 2

DHARSHANIYA V

Team Member 3

MAHALAKSHMI G

1.INTRODUCTION

1.1Project Overview

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.

1.2Purpose :

Newspapers are one of the most popular and most needed commodities in our daily life. In today's busy world, reading newspapers has become one of the traditional ways of getting news. News is produced every minute and distributed via television, radio and the Internet, so the news updated the next morning is already outdated. So newspaper and magazine publishers have a hard time keeping up with the pace. Change is needed and publishers must embrace mobile.

2.LITERATURE SURVEY

(PDF) Exploring mobile news reading interactions for news app personalisation (researchgate.net)

(PDF) Topic Detection and Tracking in News Articles (researchgate.net)

(PDF) Following the Fed with a News Tracker (researchgate.net)

(PDF) Following the Fed with a News Tracker (researchgate.net)

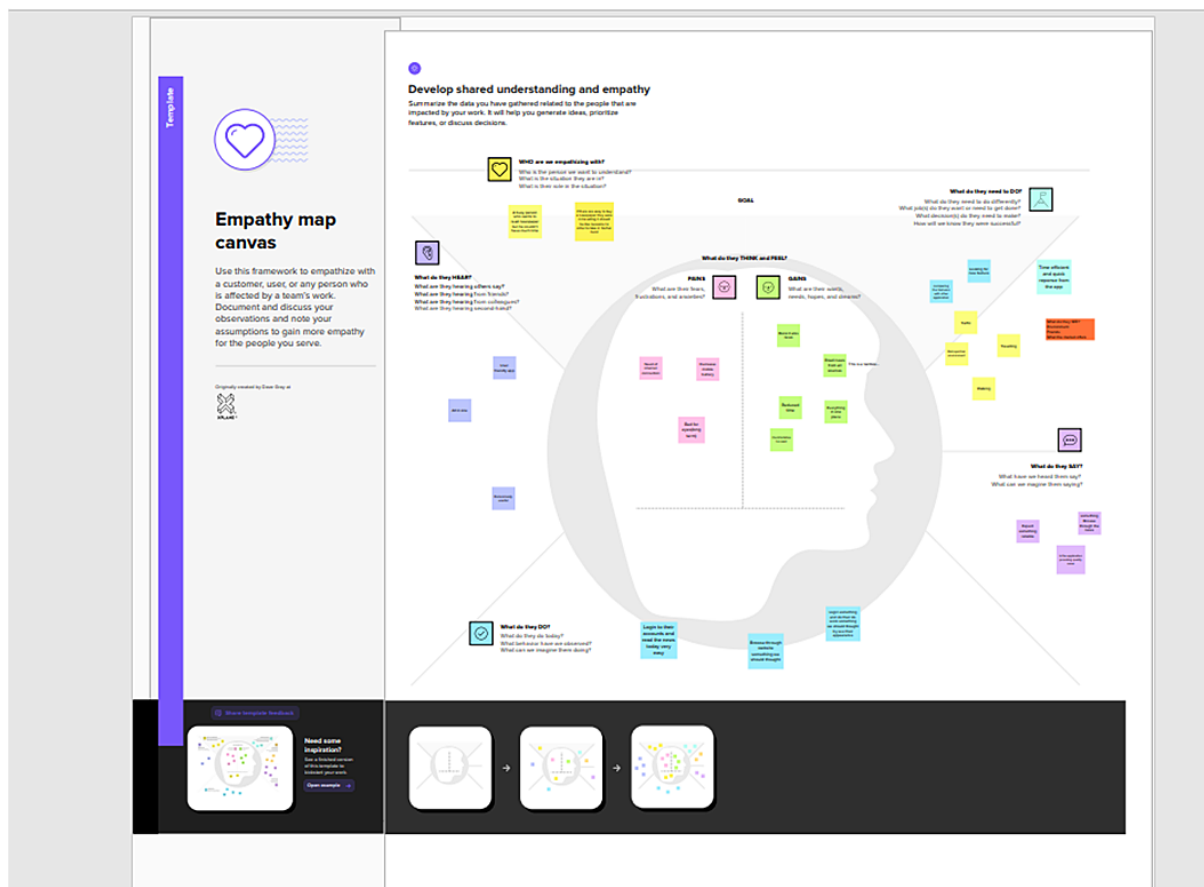
Reference:

- 1.<https://flask.palletsprojects.com/en/2.1.x/tutorial/>
- 2.<https://www.ibm.com/docs/en/db2-for-zos/11?topic=zos-db2-basics-tutorial-working-db2-interactively>
- 3.<https://www.ibm.com/in-en/products/watson-assistant/docs-resources>
- 4.<https://www.ibm.com/blogs/cloud-archive/2019/03/video-containerization-explained/>
- 5.<https://developer.ibm.com/components/cloud-foundry/tutorials/>

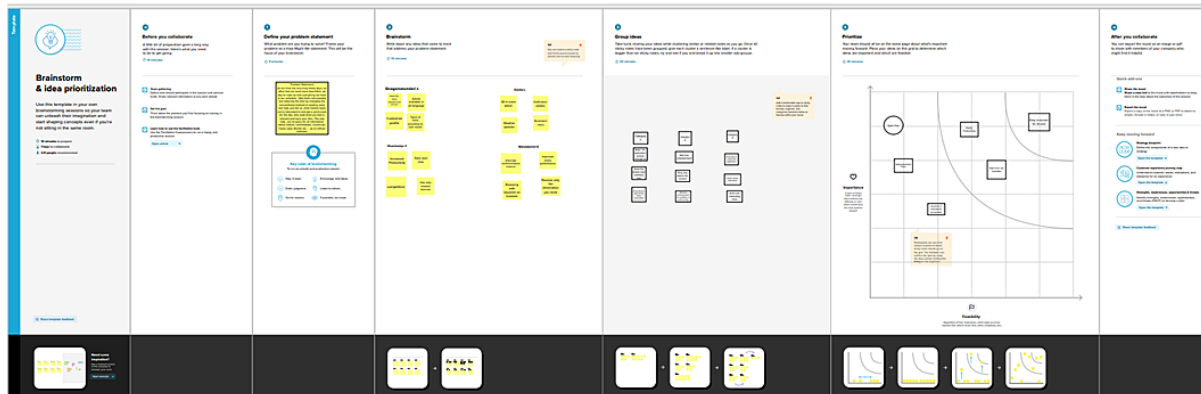
2.2Problem Statement Definition

In today's world we have a lot of work and there is no time to cope up everything in the schedule. So, it is not possible to read the newspaper. By using this application, the user can access the news they are interested and get a quick peek for the day. 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 newsssharing app wants to help users find relevant and important news easily every day andalso understand explicitly that the news is not fake but from proper sources. This news tracker application helps the user to get all information about commodities, sports, technology, education etc. The user can register using their

3.1. Empathy Map Canvas



3.2.Ideation & Brainstorming



3.3.Proposed Solution

S.NO	Parameter	Description
1	Problem Statement (Problem to be solved)	Statement: Everyday, a lot of events happen world-wide and we rely on newspapers, television and news articles to get the reliable and trust-worthy information about these events. Description: As a result, we created a platform that offers such news from reliable sources worldwide, in an organized and efficient manner.
2	Idea / Solution description	One platform for all local and worldwide news. Trustworthy and Reliable News. Fast and efficient system. Preventing spread of False information. Data Storage and Backup. Communication.
3	Novelty / Uniqueness	A cloud computing-based news application that generates news and reports about the happenings around the world using computers and network (Internet). News based on most reliable and trustworthy resources around the world. Developing the Eco- Friendly &

		sustainability based on centre.
4	Social Impact/ Customer Satisfaction	Cloud computing offers a way to create, coordinate, and share information across the globe. The adoption of cloud-based services gives access to a wider range of data and sharing the important
5	Business Model (Revenue mode1)	Building mobile tools with data isn't are easy as importing an xml feed of your latest headlines. But id you're going to spend thousands of dollars developing a mobile app anyway. You canmight as well spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problem and makes advertisers take notices
6	Scalability of the solution	News tracker application can handle data across machines and data that will not fit into memory. It supports clusters and can handle machine failures, rebuilding machine easily.

Problem solution fit:

Define CS,fit into CC 1.Customer Segment(S) Youngsters,Understudie s, Old Matured People.	6.Customer Constraints Time Utilization,SavesTime,Ne ed of Website,Effective of Cell phone or laptop,simple to utilize,Effectively Justifiable by Everybody	Explore As, differentiate 5.Available Solution Cell phones can be accused of less current instesd of TV and radio's,helpful to utilize and can be effectively conveyed to all over.
Focus on J&P,tap into BE,understand RC 2.Jobs to be Done/Problems Different Perspective on the Client and their Fullfillment.	9.Problem Root Cause Client can introduce this application to save their time and simple to utilize.They no need of TVs or Radios to convey any place they need.Just they need a cell phone with web office	Focus on J&P,tap into BE, understand RC 7.Behaviour User needs to install this application from a verified server. Needs Internet Facility throughout this application and to read news.

<p>Identify Strong TR&EM</p> <p>3.Triggers My environmental elements has been introduced this application and I cherished it to utilize in light of the fact that,It save my time</p> <p>4.Emotion:Before/After Viewed Only at home>anywhere at any time-this application is useful and can be used whenever we want</p>	<p>10.Your Solution In this News Following applications,A portion of the news were phony and a portion of the news was genuine and clients might get irritated due to this application and they could tell to this environmental factor do the impression of the application could get down.To determine this one an administrator Bot is made and at whatever point News Get Refreshed in this application.This bot will actually look at through web and assuming that it is phony the bot naturally.</p>	<p>Extract online&Offline CH&BE</p> <p>8.Channels of behaviour 8.1 Online: Client can do everything in this application utilizing on the web(web)</p> <p>8.2 Offline: Client can download significant news when the client has web and when the client is disconnected.They can view the Downloaded.</p>
--	--	--

4. Requirement Analysis

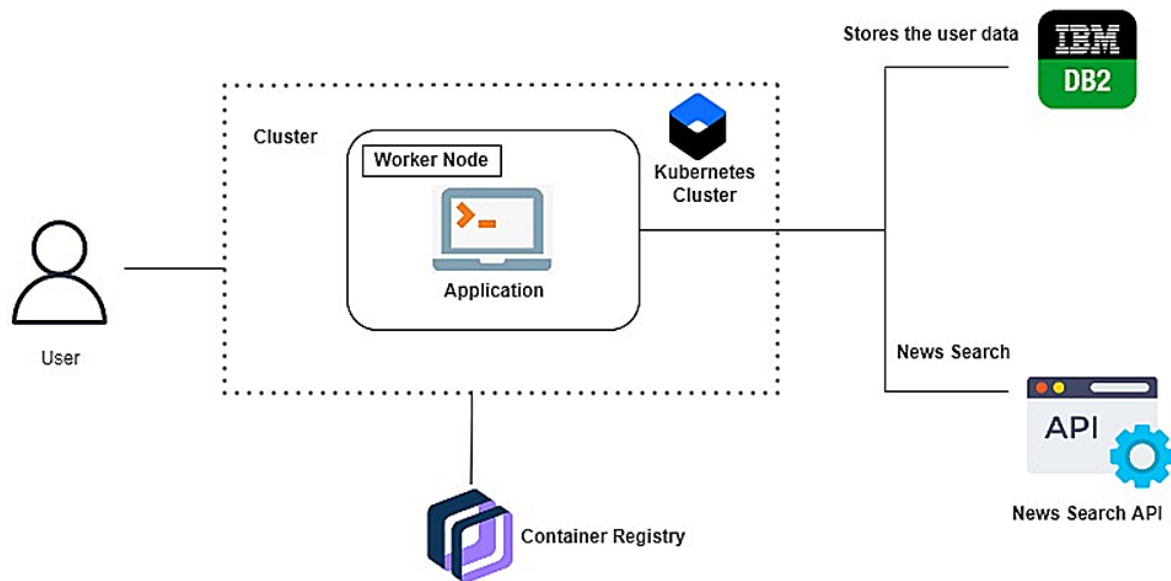
4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Installation	User can install the app from Google play store or from the website
FR-2	User Registration	Registration through Form Registration through Gmail
FR-3	User Confirmation	Confirmation via Email
FR-4	User Login	User should login the app with the user's name and password

4.2 Non-Functional requirements

FR No	Non-Functional Requirement	Description
NFR-1	Usability	Everyone can understand the process of using the app easily by the commands given in the app
NFR-2	Security	It is a more secured app. No fake news can be shared.
NFR-3	Performance	Performance of the app is very great
NFR-4	Availability	More sub categories are available

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

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

5.3 User Stories

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.

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 mahaDevi my password.	2	High	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-1		USN-2	s a user, I will receive confirmation email once I have registered for the application	1	High	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Sivagamasudari Devika L Mahalakshmi Dharshaniya

Sprint=1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-3	Dashboard	USN-6	As a user, I will able to see the login and profile details	2	Medium	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-1	IBM cloud	USN-10	As a tracker, I collect all the NEWS from the cloud	1	Medium	Sivagamasudari Devika L Mahalakshmi Dharshaniya
Sprint-3	IBM cloudant	USN-11	As a database manager, I can collect all the user details and NEWS application details	2	Medium	Sivagamasudari Devika L Mahalakshmi Dharshaniya

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3. Reports from JIRA

Sprint-1

The screenshot displays the Jira Software interface for a project named 'my testings'. The left sidebar shows navigation options: PLANNING (Roadmap, Backlog, Board), DEVELOPMENT (Code), Project pages, Add shortcut, and Project settings. The main area is titled 'Backlog' and shows a list of issues for 'ITT Sprint 1 15 Nov - 25 Dec (5 issues)'. The issues are:

- hg (Issue ID: ITT-4)
- creating flask project (Issue ID: ITT-9)
- creating database (Issue ID: ITT-12)
- ideation phase (Issue ID: ITT-11)
- deployment of app in cloud (Issue ID: ITT-10)

All issues are marked as 'DONE' and assigned to team members. A 'Quickstart' button is visible at the bottom right. The Windows taskbar at the bottom shows the date as 15-11-2022 and time as 11:29 AM.

Sprint-2

my testings Software project

PLANNING

- Roadmap
- Backlog
- Board

DEVELOPMENT

- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / my testings

Backlog

Search

ITT Sprint 2 15 Nov – 11 Dec (4 issues)

- ibm_db python package DONE
- ibm cloudant package DONE
- ibm cloudant package creating login page DONE
- creating sign in page DONE

+ Create issue

ITT Sprint 1 11 Dec – 25 Dec (4 issues)

ITT Sprint 3 25 Dec – 8 Jan (4 issues)

Quickstart

Sprint-3

my testings Software project

PLANNING

- Roadmap
- Backlog
- Board

DEVELOPMENT

- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / my testings

Backlog

Search

ITT Sprint 3 15 Nov – 8 Jan (4 issues)

- backend code of login page DONE
- change the data in json format to store in database DONE
- connecting the ui interface with database DONE
- backend code for sign in page DONE

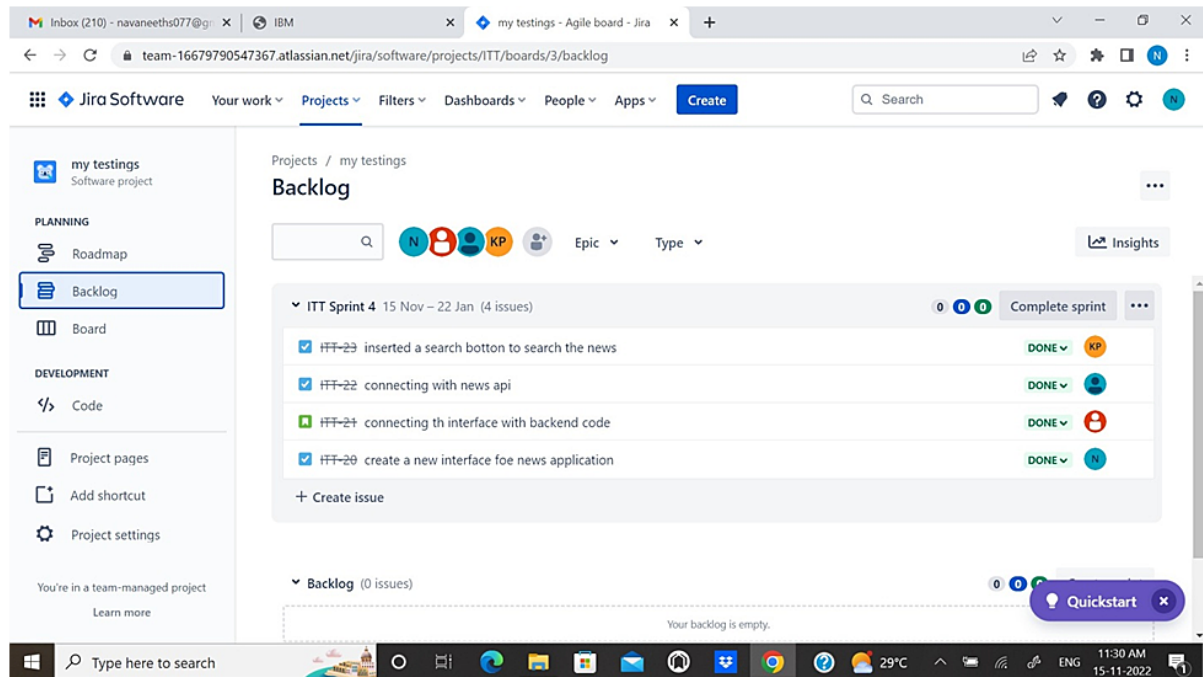
+ Create issue

ITT Sprint 4 8 Jan – 22 Jan (4 issues)

- create a new interface for news application IN PROGRESS

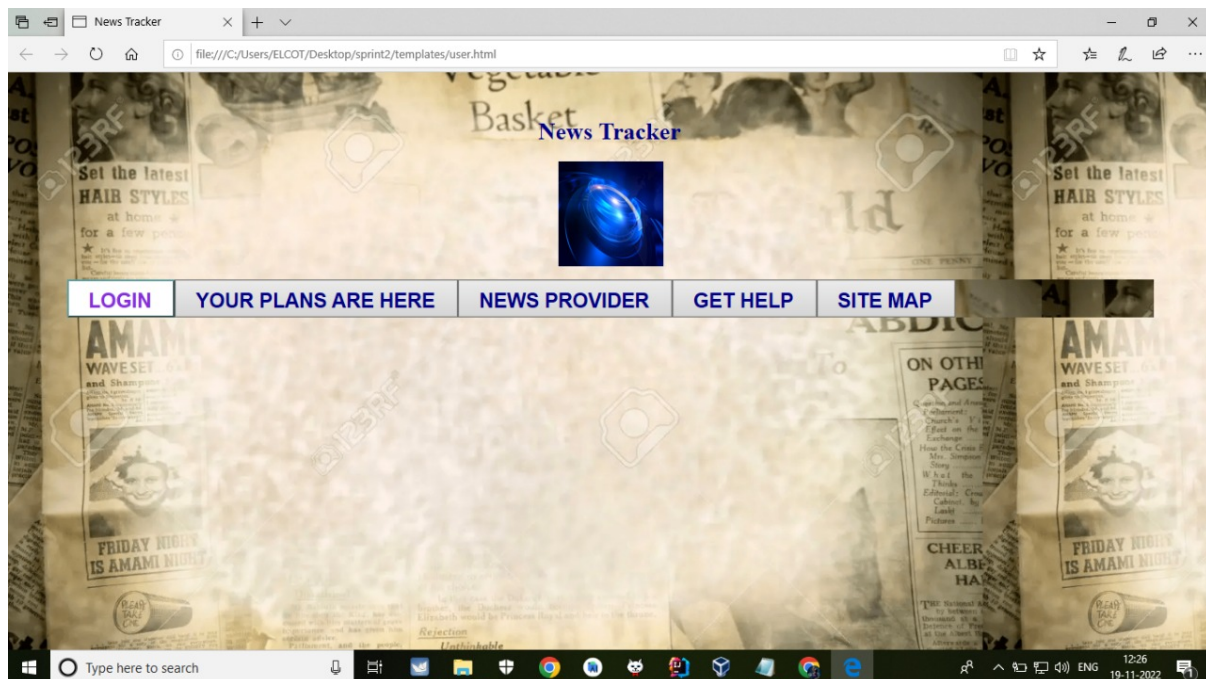
Quickstart

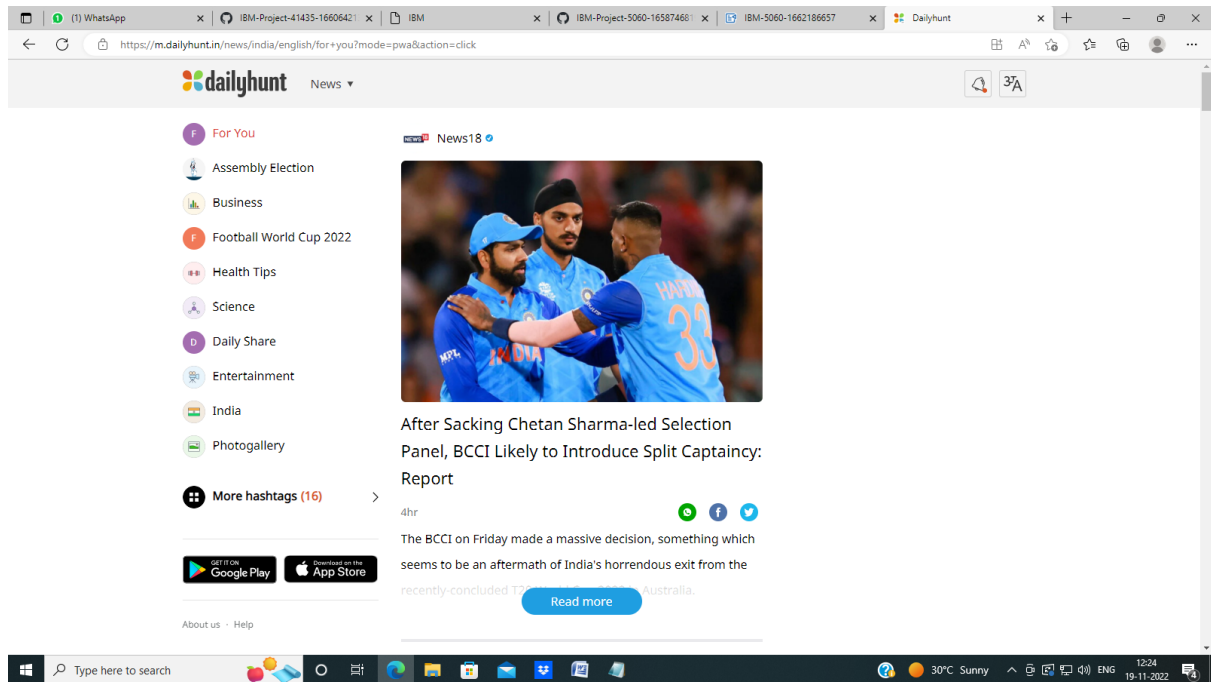
Sprint-4



7. CODING & SOLUTIONING

7.1 Feature 1





8. TESTING

8.1. User Acceptance Testing

Acceptance Testing

UAT Execution & Report submission

Date	03 November 2022
Team ID	PNT2022TMID49391
Project Name	News Tracker application
Maximum marks	4 marks

1. Purpose of the project

The purpose of the document is to briefly explain the test coverage and open issues of the [product Name] project at the time of the release to the User Acceptance Testing

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity

Resoulution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

This report shows the number of test cases that have passed, failed and untested

Section	Total cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception responding	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8. TESTING

8.1. User Acceptance Testing

Test case ID	Feature Type	Component	Test Scenario	Pre-Requlite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Automation (T/N)	Tester ID	Executed By
egghspg_TC_001	Functional	Home Page	Verify user is able to see the Login/Register popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Register popup displayed or not	http://www.egghspg.com/	Login/Register popup should display	Working as expected	Pass				Emppulish
egghspg_TC_002	UI	Home Page	Verify the UI elements in Login/Register popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Register popup with below UI elements: a)Email text box b)password text box c>Login button d)New customer? Create account link e)Lost password? Recovery password link	http://www.egghspg.com/	Application should show below UI elements: a)Email text box b)password text box c>Login button with orange colour d)New customer? Create account link e)Lost password? Recovery password link	Working as expected	Fail	Steps are not clear to follow		BUG-1234	Kevin D'Souza
egghspg_TC_003	Functional	Home Page	Verify user is able to log into application with Valid credentials		1.Enter URL(http://www.egghspg.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: dsham@gmail.com password: Turling123	User should be redirected to user account homepage						Monarch
egghspg_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(http://www.egghspg.com/) and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: dsham@gmail.com password: Turling123	Application should show "Invalid email or password" validation message.						Monarch Raju
egghspg_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(http://www.egghspg.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: dsham@gmail.com password: Turling1234567890123456789	Application should show "Invalid email or password" validation message.						Monarch
egghspg_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL(http://www.egghspg.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: dsham@gmail.com password: Turling1234567890123456789	Application should show "Invalid email or password" validation message.						Monarch Raju

9.RESULTS

9.1Performance Metrics

The image displays two screenshots of a web application interface, likely for a news tracker, showing the login and registration forms.

Top Screenshot: LOGIN

The login form is titled "LOGIN" and is located in the center of the page. It contains the following fields and buttons:

- USERNAME:
- PASSWORD:
- SUBMIT:

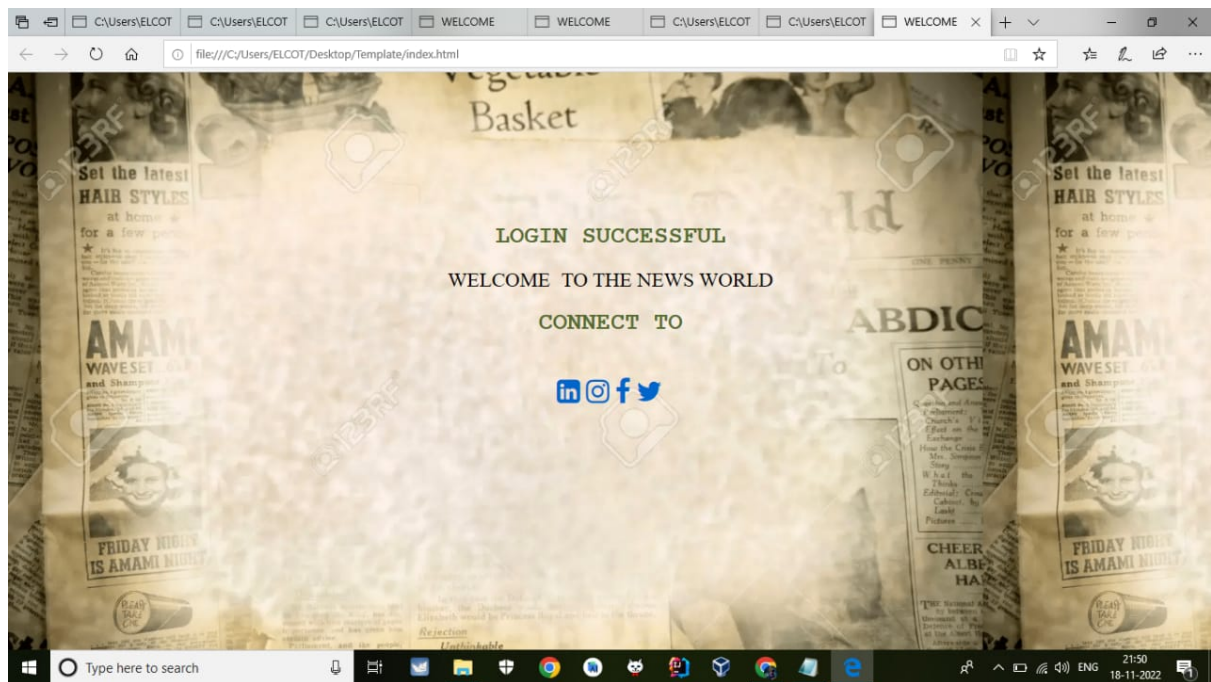
Below the login form, there is a link for "New User" and a "SIGN UP" button.

Bottom Screenshot: REGISTRATION FORM

The registration form is titled "REGISTRATION FORM" and is located in the center of the page. It contains the following fields and buttons:

- USERNAME:
- EMAIL ID:
- PHONE NUMBER:
- PASSWORD:
- Submit:

Both screenshots show a background image of a newspaper clipping with various headlines, including "Vegetable Basket", "Set the latest HAIR STYLES at home for a few pennies", "AMAMI WAVESET and Shampoo", "FRIDAY NIGHT IS AMAMI NIGHT", "ABDIC", "ON OTHER PAGES", and "CHEER ALBE HA".



10. ADVANTAGES

1. All trending news will be displayed in an orderly manner.
2. We can customize the topic which we are interested and make them display The user experience are quite simple and friendly

10. DISADVANTAGES

1. Same news which is published in different websites can be displayed.
2. Delay response of news api can make the news collapse.

11. CONCLUSION

Thus the projected using IBM CLOUD is tested, verified and executed successfully.

12. Future scope

In future, we planned to implement this project in large scale which will be helpful and used by all the people.

13.APPENDIX

Source Code

Login.html

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
  background-image:url("news1.png");
  background-color:lightgray;
}
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<style>
  body {
    background-image: linear-gradient(92.7deg, rgb(243, 178, 178) 8.5%, rgb(246, 244, 198)
90.2%);
    font-family: Georgia, 'Times New Roman', Times, serif;
  }

  button:hover {
    background-color: dark blue;
    border-color: black;
  }

  h1 {
    font-family: 'Courier New', Courier, monospace;
    color: black;
    top: 10em;
  }

  .container1 {
    border: 2px solid black;
    border-color: black;
    border-radius: 10px;
```

```

        width: 400px;
    }

    .top {
        margin-top: 100px;
    }

    input:hover {
        border-color: rgb(25, 20, 20);
    }

    a {
        text-decoration: none;
    }

    a:link {
        color: #006600;
        text-decoration: none;
    }

    a:visited {
        color: rgb(215, 117, 92);
        text-decoration: none;
    }

    a:hover {
        color: rgb(128, 105, 255);
        text-decoration: none;
    }

    a:active {
        color: rgb(202, 99, 75);
        text-decoration: none;
    }
</style>

<body>
    <center>
        <h1 class="top">NEWS TRACKER</h1>
        <div class="container1">
            <br>
            <h1>LOGIN</h1>
            <form action="http://localhost:5000/login" method="POST">
                <table>

```

```

        <tr>
            <td><label for="text">USERNAME</label></td>
            <td><input type="text" name="username" placeholder="enter user name"
/></td>
        </tr>
        <tr>
            <td><label for="text">PASSWORD</label></td>
            <td><input type="text" name="password" placeholder="enter valid
password"></td>
        </tr>
    </table>
    <br>
    <button type="submit">SUMBIT</button>
</form>
<br>
</div>
<br>
<label for="text">New User <br> <br> <b><a href="http://http://localhost:5000">SIGN
UP</a></b></label>
</center>
</body>

</html>

```

Register.html

```

<!DOCTYPE html>
<html>
<head>
<style>
body{
    background-image:url("news1.png");
    background-color:lightgray;
}
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>REGISTRATION PAGE</title>
</head>
<style>

```

```

body {
    background-image: linear-gradient(92.7deg, rgb(59, 170, 201) 8.5%, rgb(246, 244, 198)
90.2%);
    font-family: 'Times New Roman', Times, serif;
}

input:hover {
    border-color: rgb(25, 20, 20);
}

button:hover {
    background-color: SpringGreen;
    border-color: black;
}

h1 {
    font-family: 'Courier New', Courier, monospace;
    color: darkolivegreen;
}

#qwerty {
    margin-top: 15em;
}
</style>

```

```

<body>
    <center id="qwerty">
        <H1>REGISTRATION FORM</H1>
        <!-- -->
        <form action="http://localhost:5000/register" method="POST">
            <table>
                <tr>
                    <td><label for="text">USERNAME</label></td>
                    <td>&nbsp;</td>
                    <td><input type="text" placeholder="Enter Username" name="username"
id="username"></td>
                </tr>
                <tr></tr>
                <tr></tr>
                <tr>
                    <td><label for="text">EMAIL ID</label></td>
                    <td>&nbsp;</td>
                    <td><input type="text" placeholder="Enter email id" name="email_id"
id="email_id"></td>

```



```

        </tr>
        <tr></tr>
        <tr></tr>
        <tr>
            <td><label for="text">PHONE NUMBER</label></td>
            <td>&nbsp;</td>
            <td><input type="text" placeholder="Enter phone number" name="phone_no"
id="phone_no"></td>
        </tr>
        <tr></tr>
        <tr></tr>
        <tr>
            <td><label for="text">PASSWORD</label></td>
            <td>&nbsp;</td>
            <td><input type="text" placeholder="Enter password" name="password"
id="password"></td>
        </tr>

        <tr></tr>
        <tr></tr>
    </table>
    <br>
    <center><button onclick="asd()" type="submit">Submit</button>
    </center>
</form>
</center>
</body>
<script>
    function asd() {
        var username1 = document.getElementById("username");
        var email_id = document.getElementById('email_id');
        var phone_no = document.getElementById('phone_no');
        var password = document.getElementById('password');
        if (username1.value == "" || phone_no.value == "" || password.value == "") {
            username.style.borderColor = "red";
        }
        else if (email_id.value == "") {
            email_id.style.borderColor = "red";
        }
        else if (phone_no.value == "") {
            phone_no.style.borderColor = "red";
        }
        else if (password.value == "") {
            password.style.borderColor = "red";
        }
    }

```

```
}  
  
}  
</script>  
</html>
```

Index.html

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body{  
  background-image:url("news1.png");  
  background-color:lightgray;  
}  
</style>  
</head>  
<body>  
<html lang="en">  
  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/4.7.0/css/font-awesome.min.css">  
  
  <title>WELCOME</title>  
</head>  
<style>  
  h1 {  
    font-family: 'Courier New', Courier, monospace;  
    color: darkolivegreen;  
    top: 10em;  
  }  
  
  body {  
    background image:  
    url ('image.jpg');  
  
  }
```

```
a:hover {
    color: cmy(200,170,70);
    text-decoration: none;
}
```

```
.font {
    color: cmy(141, 18, 100);
    font: bold;
    font-size: 27px;
}
```

```
#top {
    margin-top: 200px;
}
```

</style>

<body>

<center>

<H1 id="top">LOGIN SUCCESSFUL</H1>

<table>

<tr class="font">

<td>WELCOME</td>

<td> </td>

<td>TO THE NEWS WORLD</td>

</tr>

<!-- <tr class="font">

<td>EMAIL_ID</td>

<td> </td>

<td>{{email_id}}</td>

</tr>

<tr class="font">

<td>PHONE NO</td>

<td> </td>

<td>{{phone_no}}</td> -->

<!-- </tr> -->

</table>

<table>

<tr>

<h1>CONNECT TO</h1>

</tr>

<tr>

<pre class="tab">

<td>

```

        <label"><a href="https://in.linkedin.com/company/ibm"><i class="fa fa-linkedin-
square"
        style="font-size:36px"></i></a></label>&nbsp;
    </td>

    <td>

        <a href="https://www.instagram.com/ibm/?hl=en"><i class="fa fa-instagram"
        style="font-size:36px"></i></a>&nbsp;
    </td>

    <td>

        <a href="https://www.facebook.com/IBM/"><i class="fa fa-facebook" style="font-
size:36px"></i></a>&nbsp;
    </td>
    <td>

        <a href="https://twitter.com/ibm?lang=en"><i class="fa fa-twitter" style="font-
size:36px"></i></a>&nbsp;
    </td>
</pre>
</tr>
</table>

</table>
</center>

```

```
</body>
```

```
</html>
```

```
</body>
```

```
</html>
```

app.py

```

from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)
hostname = '19af6446-6171-4641-

```

```

8aba9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'sfr69790'
pwd = 'VFRS0bFKFelhMbcq'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '30699'
protocol = 'TCPIP'
cert = "C:/Users/Kaviyarasu/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('/', methods=['GET', 'POST'])
@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = " "
    if request.method == 'POST':
        username = request.form['username']
        email_id = request.form['email_id']
        phone_no = request.form['phone_no']
        password = request.form['password']
        query = "SELECT * FROM USER1 WHERE username=?;"
        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 USER1 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, phone_no)
ibm_db.bind_param(stmt, 4, 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('/login', methods=['GET', 'POST'])
def login():
global userid
msg = ''
if request.method == "POST":
username = request.form['username']
password = request.form['password']
query = "select * from user1 where username=? and password=?"
stmt = ibm_db.prepare(connection, query)
ibm_db.bind_param(stmt, 1, username)
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['USERNAME']
session['username'] = account['USERNAME']
msg = 'Logged in Successfully'
return render_template('welcome.html', msg=msg, username=str.upper(username))
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('welcome.html', username=username)
else:
    return render_template('welcome.html', username=username)
if __name__ == "__main__":
    app.run(debug=True)
    app.run(host='0.0.0.0')

```

Verify.css

```

@import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@100;300&display=swap');

```

```

*,*::before,*::after{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Roboto', sans-serif;
}
body{
    background-color: aliceblue;
    min-height: 100vh;
}
nav {
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    align-items: center;
    margin: auto;
    min-width: 100%;
    box-shadow: 0 0 5px rgba(35, 35, 35, 0.584);
    background-color: #d9d9d9be;
    padding: 1rem;
}

```

```

.status-cont{
    display: flex;
    min-height: 70vh;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    font-weight: medium;
    text-align: center;
}

```

```
padding: 1rem;
}

.welcome-text{
  font-size: 2.5rem;
}

.sub-text{
  font-size: 1.5rem;
}
```

GitHub :

<https://github.com/IBM-EPBL/IBM-Project-5060-1658746819>

Video link:

https://drive.google.com/file/d/1Y3BgOoIVcFr-7D33ngnXH52_8A8xqRL3/view?usp=drivesdk

