PROJECT REPORT

TEAM ID   : PNT2022TMID39497
PROJECT : NEWS TRACKER APPLICATION

1. INTRODUCTION

1.1 Project Overview

News Tracker is a full stack web application which allows users to register along with their favourite topics, upon login the app displays the news based on the user's interest. The news displayed in the app is based on the New catcher API and Cric buzz API from Rapid API site. A news-sharing app wants to help users find relevant and important news easily every day and also provide explicitly news from that users locality/region which may of help to the user.

1.2 Purpose

Enabling users to view news from anywhere at anytime. It also helps to reduce the time to get information about a specific topic. Also enables a person to get an updated news which may help Business people to make business related decisions quickly and correctly.

2. LITERATURE SURVEY

2.1 Existing problem

A news application is a major intelligent data set that recounts a report. Think about it like you would some other piece of news coverage. It simply utilizes programming rather than words and pictures. As our lives are exceptionally bustling nowadays, we frequently feel we want in excess of 24 hrs. a day to adapt up to all that we have in our timetable. Indeed, that is unrealistic yet diminishing the time by changing the traditional strategy for pursuing news can help. Simply let us know what market news you're keen on and get a fast look for the afternoon. Just read what you feel is applicable and save your time. This application assists you with questioning all data about Indices, Commodities, Currencies, Future Rates, Bonds, and so on as on true sites.

INTRODUCTION:

The client wanted to create a solution to facilitate work for media professionals through a location-based assignment management tool. The main challenge was to create a system on which journalists and other staff remained connected, and could locate or contact each other swiftly to relay stories faster. The built a News Tracker to collect a variety of sources of misinformation on Facebook and aggregate the stories published so that they could have a clearer sense of the different types of misinformation out there. They wanted to know how frequently it was published and what strategies and narratives were employed to engage audiences on Facebook.

LITEREATURE SURVEY:

Author Describe [1], The Europe Media Monitor system (EMM) gathers and aggregates an average of 50,000 newspaper articles per day in over 40 languages. To manage the information overflow, it was decided to group similar articles per day and per language into clusters and to link daily clusters over time into stories. Large amounts of information are published daily on news web portals around the world. Presenting the most important news on simple, newspaper-like pagesis enough when the user wants to be informed about the latest news. Author Describe [2], The key findings of these three studies are: (1) mobile new readers can be characterized within three types; (2) it is possible to detect a user's newsreader type from analysis of their interactions with a news app, and; (3) different reader types benefit from different forms of news app interface. We showed that mobile news readers can be distinguished as Trackers, Reviewers or Dippers according to their frequency of news reading ,their duration of news reading, their browsing strategy, their reading style and their location. We showed that a Bayesian classifier can identify the user of a news app according to their news reader type from a log of interactions with a news app from which the clustering factors could be extracted. The evaluation study demonstrated that Trackers performed better with the adapted interface variant, whereas Reviewers and Dippers performed better with the baseline interface.

REFERENCES:

[1]. Vehicle Tracking Systems Overview [Online:] http://www.roseindia.net/technology/vehicletracking/ VehicleTrackingSystems.shtml. [2][Pouliquen, B., Steinberger, R., & Olivier Deguernel, O. (2008, August). Story tracking: linking similar news over time and across languages. In Coling 2008: Proceedings of the workshop Multi-Source Multilingual Information Extraction and Summarization (pp. 49-56).
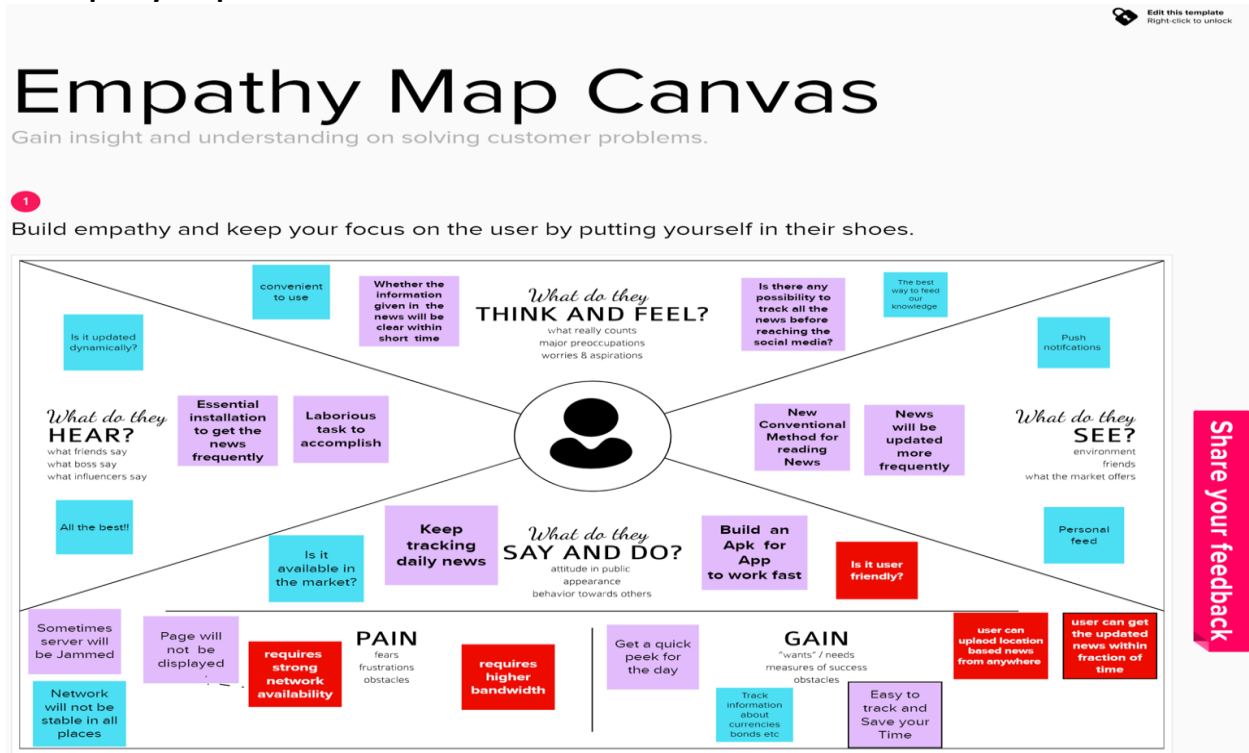
2.3 Problem Statement Definition :

PROBLEM STATEMENT:

     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.
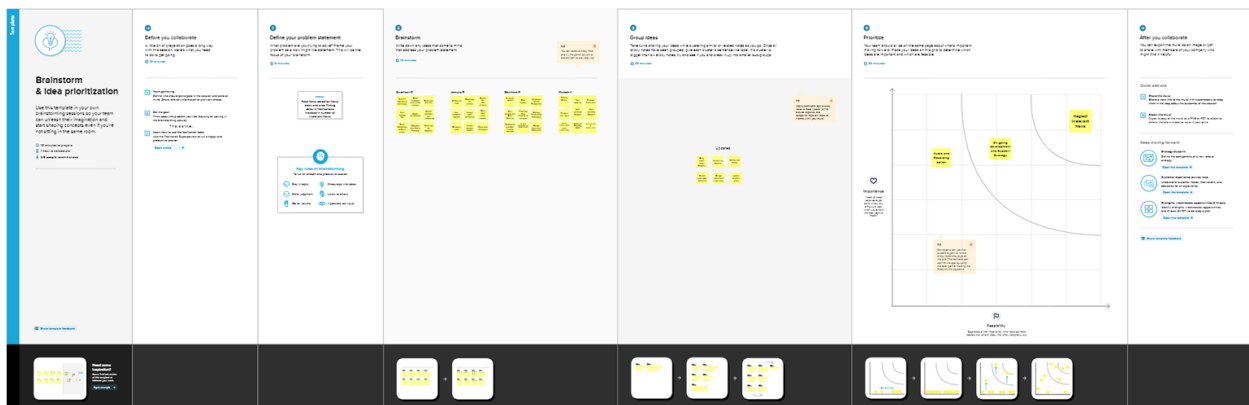
QUESTIONS	DESCRIPTION
What is the issue?	Physical newspapers are old fashioned in this digital era. They cost money to buy, can easily be damaged, limited amount of information, not flexible to modifications, poor quality. Fixing these in such a way can make physical newspapers become extinct and their use can be abolished. If the problem isn't solved, it would become a greater problem to integrate people to become digitally aware.
When does this issue occur?	The issue occurs when the customer wants to read the news.
Who does the problem affect?	Customers
What are the boundaries of the problem?	geographic, workflow, daily life
Where is the issue occuring?	The issue occurs when people try to access information i.e during the usage. Accessing information is flawed through the usage of physical newspapers in the digital era.

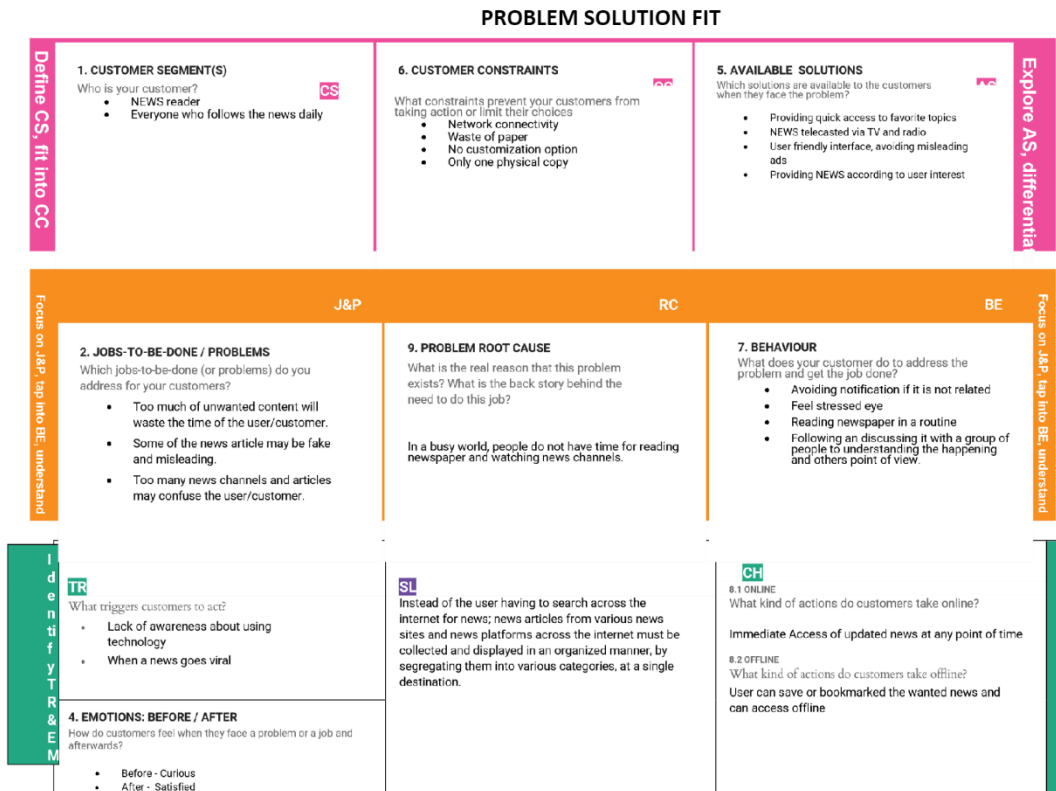# 3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation and Brainstorming

3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Fake news detection • News clean and crisp • Time delay in Notifications • Increase in number of irrelevant news
2.	Idea / Solution description	• Improve app speed and usage • Automatic speech recognition • User privacy and data security • Create own Wishlist • Authorized news • Compact with mobile platform
3.	Novelty / Uniqueness	• Fast loading and higher performance • Category news • Easy access and able to share with the colleagues • Helpful customer care support
4.	Social Impact / Customer Satisfaction	• Daily email news report • Providing more flexibility for everyone • On going development and support strategy • Neglect irrelevant news
5.	Scalability of the Solution	In scale of 10 : • As a product - 8.5 • In point of user - 8 • As a business - 9 • As a developer -9.2
6.	Business Model (Revenue Model)	• Affordable cost of subscription • Multiplatform Support • Design consistent experience • Minimization of user input and work progress

3.4 Problem Solution fit :

PROBLEM SOLUTION FIT

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
- NEWS reader
- Everyone who follows the news daily

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices
- Network connectivity
- Waste of paper
- No customization option
- Only one physical copy

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem?
- Providing quick access to favorite topics
- NEWS telecasted via TV and radio
- User friendly interface, avoiding misleading ads
- Providing NEWS according to user interest

Explore AS, differential

Focus on J&P, tap into BE, understand

J&P | RC | BE

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?
- Too much of unwanted content will waste the time of the user/customer.
- Some of the news article may be fake and misleading.
- Too many news channels and articles may confuse the user/customer.

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back story behind the need to do this job?

In a busy world, people do not have time for reading newspaper and watching news channels.

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?
- Avoiding notification if it is not related
- Feel stressed eye
- Reading newspaper in a routine
- Following an discussing it with a group of people to understanding the happening and others point of view.

Focus on J&P, tap into BE, understand

Identify TR & EM

TR

What triggers customers to act?
- Lack of awareness about using technology
- When a news goes viral

SL

Instead of the user having to search across the internet for news; news articles from various news sites and news platforms across the internet must be collected and displayed in an organized manner, by segregating them into various categories, at a single destination.

CH

8.1 ONLINE
What kind of actions do customers take online?

Immediate Access of updated news at any point of time

8.2 OFFLINE
What kind of actions do customers take offline?
User can save or bookmarked the wanted news and can access offline

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
- Before - Curious
- After - Satisfied

REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Installation	User can install the app from google play store, App store and from website.
FR-2	User Registration	Registration through Form Registration through Gmail Registration through Mobile number.
FR-3	User Confirmation	Confirmation via Email Confirmation via OTP
FR-4	User login	User should login the app with the User's name or email and password.
FR-5	User Information	User can include their information in preferred topics so that they could be recommended by the application.

Non-functional Requirements:

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

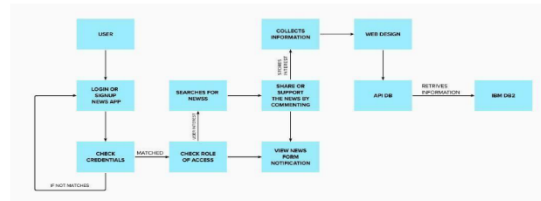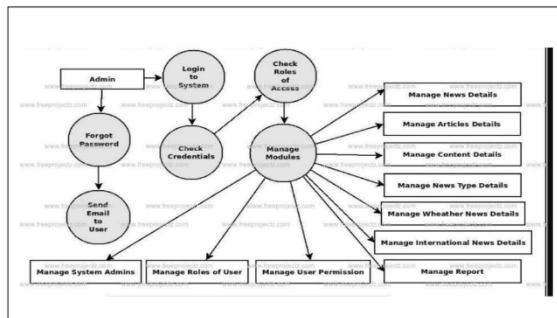| FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This app is user friendly app, where users can clear their queries with the chatbot.
NFR-2	**Security**	This app is secured app, where users information is encrypted properly.
NFR-3	**Reliability**	This app can be accessed anywhere and 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. Variety of news is provided by this app. More sub categories are available.

PROJECT DESIGN

5.1 Data Flow Diagrams:

Project Design Phase-II
Data Flow Diagram & User Stories

Date	09 November 2022
Team ID	PNT2022TMID39497
Project Name	NEWS TRACKER APPLICATION
Maximum Marks	4 Marks

Data Flow Diagrams:

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.

(DFD) for news tracker application for login:

User Stories :

USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login
		USN-4	As a user, I can register for the application through Gmail	I can receive password to mail.
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation mail.
	Dashboard	USN-6	The news portal and shows the recent news as Breaking NEWS	I can Open and view the news portal.
Customer (Web user)	Search Bar	USN-7	User searches for news based on their own interest.	I can view the related news and can watch videos.
		USN-8	The news can be viewed that is appearing on the dashboard.	On the dashboard. I shall click on the news wanted and can open it.
Administrator	Server	USN-9	Provides correct news available from the database.	Provide the correct news from the database and fake news will be rejected
		USN-10	Provide live news with video and audio contents.	I can get the news in which I am interested.

2

5.2 Solution Architecture and Technical Architecture :
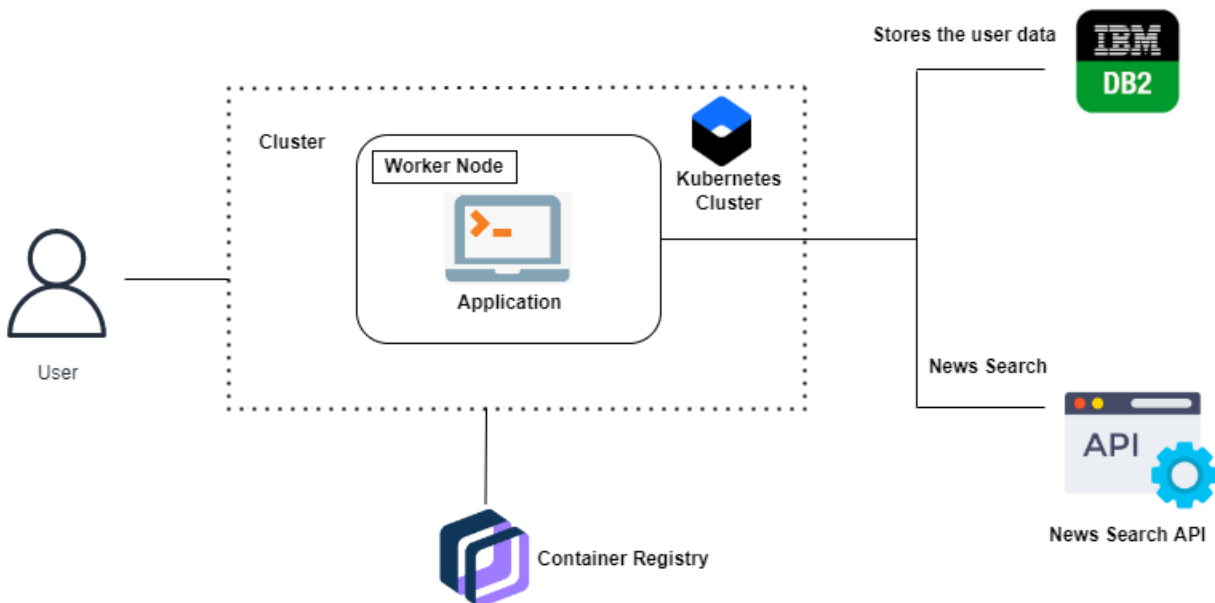
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.
Its goals are to:
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders

Technical Architecture :

Project Workflow: -

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

Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User interacts with the application using web development, which is used to get the information needed for the user.	HTML, CSS, JavaScript / React Js etc.
2.	Application Logic	The Logic depends on the extraction news from the database	Python
3.	Database	Data type , configuration etc.	MYSQL, NOSQL, etc.
4.	Cloud database	Database service on cloud	IBM DB2, IBM Cloudnet etc..
5.	File storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem.
6.	Infrastructure (server/cloud)	Application deployment on local system / cloud local server configuration : Docker Cloud server configuration : Kubernetes	Local, Cloud Foundry, Kubernetes, etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.

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 register & access the dashboard with Gmail Login	Low	Sprint-2
	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-1	As a user, I can log into the application and look into my dashboard	I can Look into My Dashboard After my login	Low	Sprint-1
		USN-2	As a user, I can log into the application and update my personal data	I can View the personal data which can be updated by the user	Low	Sprint-2
		USN-3	As a user, I can log into the application and read news based on my filter contents	News contents are filtered based on the user needs	High	Sprint-1

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation :

Product Backlog, Sprint Schedule, and Estimation

 Use the below template to create product backlog and sprint schedule

Project Planning Phase
Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	09 November 2022
Team ID	PNT2022TMID39497
Project Name	Project – NEWS Tracker Application
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)
Use the below template to create product backlog and sprint schedule

Sprint	User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sarathkumar Lokesh Jeeva Prabakaran
Sprint-1			USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Sarathkumar Jeeva
Sprint-2			USN-3	As a user, I can register for the application through Facebook	2	Low	Jeeva
Sprint-1			USN-4	As a user, I can register for the application through Gmail	2	Medium	Lokesh
Sprint-1		Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Sarathkumar Prabakaran

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	11	6 Days	25 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	3	6 Days	31 Oct 2022	05 Nov 2022	3	05 Nov 2022

Project Tracker, Velocity & Burn down Chart: (4 Marks)

Sprint-3	4	6 Days	07 Nov 2022	12 Nov 2022	4	12 Nov 2022
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 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.

Burn Down Chart

7. CODING AND SOLUTIONING

FRONTEND:

LOGIN.CSS

```css
img{
            width: 100%;
}
.login {
    height: 1000px;
    width: 100%;
    background: radial-gradient(#653d84, #332042);
    position: relative;
}
.login_box {
    width: 1050px;
    height: 600px;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%);
    background: #fff;
    border-radius: 10px;
    box-shadow: 1px 4px 22px -8px #0004;
    display: flex;
    overflow: hidden;
}
.login_box .left{
 width: 41%;
 height: 100%;
 padding: 25px 25px;

}
.login_box .right{
 width: 59%;
 height: 100%
}
.left .top_link a {
    color: #452A5A;
    font-weight: 400;
}
.left .top_link{
 height: 20px
}
```

```css
.left .contact{
        display: flex;
    align-items: center;
    justify-content: center;
    align-self: center;
    height: 100%;
    width: 73%;
    margin: auto;
}
.left h3{
 text-align: center;
 margin-bottom: 40px;
}
.left input {
    border: none;
    width: 80%;
    margin: 15px 0px;
    border-bottom: 1px solid #4f30677d;
    padding: 7px 9px;
    width: 100%;
    overflow: hidden;
    background: transparent;
    font-weight: 600;
    font-size: 14px;
}
.left{
        background: linear-gradient(-45deg, #dcd7e0, #fff);
}
.submit {
    border: none;
    padding: 15px 70px;
    border-radius: 8px;
    display: block;
    margin: auto;
    cursor: pointer;
    margin-top: 120px;
    background: #583672;
    color: #fff;
    font-weight: bold;
    -webkit-box-shadow: 0px 9px 15px -11px rgba(88,54,114,1);
    -moz-box-shadow: 0px 9px 15px -11px rgba(88,54,114,1);
    box-shadow: 0px 9px 15px -11px rgba(88,54,114,1);
}
```

```css
.right {
        background: linear-gradient(212.38deg, rgba(242, 57, 127, 0.7) 0%, rgba(175, 70, 189, 0.71) 100%),url(https://www.p
   background-size: cover;
   color: #fff;
        position: relative;
}

.right .right-text{
 height: 100%;
 position: relative;
 transform: translate(0%, 45%);
}
.right-text h2{
 display: block;
 width: 100%;
 text-align: center;
 font-size: 50px;
 font-weight: 500;
}
.right-text h5{
 display: block;
 width: 100%;
 text-align: center;
 font-size: 19px;
 font-weight: 400;
}

.right .right-inductor{
 position: absolute;
 width: 70px;
 height: 7px;
 background: #fff0;
 left: 50%;
 bottom: 70px;
 transform: translate(-50%, 0%);
}
.top_link img {
   width: 28px;
   padding-right: 7px;
   margin-top: -3px;
```

```css
        }
        .signup{
            display: flex;
            float: right;
            flex-direction: column;
            margin: 10px 10px 0px 15px;
        }
        .signup > form > button{
            border: none;
            padding: 15px 70px;
            border-radius: 8px;
            display: block;
            margin-top: 15px;
            float: right;
            background: #11daae;
            cursor: pointer;
            color: #fff;
            font-weight: bold;
            -webkit-box-shadow: 0px 9px 15px -11px rgba(242, 57, 127, 0.7);
            -moz-box-shadow: 0px 9px 15px -11px rgba(175, 70, 189, 0.71);
            box-shadow: 0px 9px 15px -11px rgba(242, 57, 127, 0.7);
        }
        .signup > form > h5{
            font-weight: bold;
            float: left;
            color: white;
            margin-right: 15px;
        }
```

NEWS.HTML

```html
<!DOCTYPE html>
<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>
    <link rel="Stylesheet" href="login.css">
</head>
<body>

        <section class="login">
```

```html
                        <div class="signup">
                            <form action="">
                    <h5>Click here to create a new Account</h5>
                    <button formaction="../index.html">SignUp</button>
                            </form>
                            </div>
                            <div class="login_box">
                                <div class="left">
                                    <div class="top_link"><a href="#"><img src="https://drive.google.com/u/0/u
                                    <div class="contact">
                                        <form action="">
                                            <h3>SIGN IN</h3>
                                            <input type="text" placeholder="USERNAME">
                                            <input type="password" placeholder="PASSWORD">
                                            <button class="submit" formaction="./news.html">LET'S G
                                        </form>
                                    </div>
                                </div>
                                <div class="right">
                                    <div class="right-text">
                                        <h2>News Tracker</h2>
                                        <h5>Login to know about what is happening??</h5>
                                    </div>
                                </div>

                            </div>
                        </section>
                    </body>
                    </html>
```

STYLE.CSS
```css
@import url("https://fonts.googleapis.com/css2?family=Roboto&display=swap");

  * {
        box-sizing: border-box;
        }
        body {
         display: absolute;
         justify-content: center;
         align-items: center;
         margin: 0;
         background-color: #f7f8fc;
```

```css
    font-family: "Roboto", sans-serif;
    color: #10182f;
}
.logo{
    display: flex;
    float: left;
}
nav{
    display: flex;
    align-items: center;
    justify-content: space-between;
}
ul{
    display: flex;
}
li{
    list-style: none;
    margin: 1rem;
    margin-right: 25px;
    margin-left: 25px ;
}

a{
    text-decoration: none;
    color: rgb(49, 49, 47);
}
a:hover {
            text-decoration: none;
            color: #daa800;
            opacity: 0.8;
            transition: 1s;
}
.container {
    display: flex;
    width: 1040px;
    justify-content: space-evenly;
    flex-wrap: wrap;
}
.card {
    margin: 10px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 2px 20px rgba(0, 0, 0, 0.2);
```

```css
  overflow: hidden;
  width: 300px;
  height: 460px;
}
.card-header img {
  width: 100%;
  height: 200px;
  object-fit: cover;
}
.card-body {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: flex-start;
  padding: 20px;
  min-height: 250px;
}

.tag {
  background: #cccccc;
  border-radius: 50px;
  font-size: 12px;
  margin: 0;
  color: #fff;
  padding: 2px 10px;
  text-transform: uppercase;
  cursor: pointer;
}
.tag-teal {
  background-color: #47bcd4;
}
.tag-purple {
  background-color: #5e76bf;
}
.tag-pink {
  background-color: #cd5b9f;
}
.tag-cyan{
  background-color: #62e2f3;
}
.tag-aas{
  background-color: #838485;
}
```

```css
.card-body p {
  font-size: 13px;
  margin: 0 0 40px;
}
.user {
  display: flex;
  margin-top: auto;
  margin-right: 20px;
  margin-top: 5px;
}

.user img {
  border-radius: 50%;
  width: 40px;
  height: 40px;
  margin-right: 10px;
}
.user-info h5 {
  margin: 0;
  margin-top: 5px;
}
.user-info small {
  color: black;
}
```

PAGE.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-Ap
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integ
<title>Daily Expense</title>
  <style type="text/css">
    body{
      background-color: #EAF6F6;
    }
    form{
```

```html
        margin-left: 550px;
        width: 30%;
    }
 </style>
</head>
<body>
    <nav class="h1 navbar navbar-inverse navbar-light p-3 mb-2 bg-warning text-light justify-content-between">
        <div class="container-fluid">
            <div class="navbar-header">
    <a class="navbar-brand" href="a">Personal Expense Tracker</a>
  </div>
 <ul class="nav navbar-nav navbar-right">
    <li><a class="navbar-brand" href="{{url_for('login')}}"><span class="glyphicon glyphicon-user"></span>Login<
    <li><a class="navbar-brand" href="{{url_for('register')}}"><span class="glyphicon glyphicon-user"></span>Reg
  </ul>
 </div>
</nav>
 <br>
  <blockquote class="blockquote text-center">
   <p class="mb-0">Fill your daily expense and minimize your unwanted expenses.</p>
   <footer class="blockquote-footer">Save and Invest your money to profit.<cite title="Job_Finder"></cite></foot
  </blockquote>
  <div class="card">
    <div class="card-body">
     Welcome Dinesh.
    </div>
   </div>
 <br>
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <h3>Add Expense</h3>
      <form action="/addexpense" method="POST">
 <div class="form-group">
            <label for="">Date</label>
            <input class="form-control" type="datetime-local" name="date" id="date">
            </div>

<div class="form-group"> <label for="">Expense name</label>
            <input class="form-control" type="text" name="expensename" id="expensename">
          </div>
  <div class="form-group">
            <label for="">Expense Amount</label>
```

```html
                    <input class="form-control"  type="number" min="0" name="amount" id="amount">
                </div>
                <div class="form-group">
                    <label for=""></label>
                    <select class="form-control" name="paymode" id="paymode">
                        <option  selected hidden>Pay-Mode</option>
                        <option name="cash" value="cash">cash</option>
                        <option name="debitcard" value="debitcard">debitcard</option>
                        <option name="creditcard" value="creditcard">creditcard</option>
                        <option name="epayment" value="epayment">UPI</option>
                        <option name="onlinebanking" value="onlinebanking">onlinebanking</option>

                    </select>

                <div class="form-group">
                    <label for=""></label>
                    <select class="form-control" name="category" id="category">
                        <option selected hidden>Category</option>
                        <option   name = "food" value="food">food</option>
                        <option name = "entertainment" value="entertainment">Entertainment</option>
                        <option name = "business" value="business">Business</option>
                        <option name ="rent" value="rent">Rent</option>
                        <option name = "EMI" value="EMI">EMI</option>
                        <option name = "other" value="other">other</option>
                    </select>
                </div>

            <input class="btn btn-danger" type="submit" value="Add" id="">

            </form>
        </div>
      </div>
    </div>
    </body>
    </html>
```

BACKEND:

NEWSTRACKER.JS

```javascript
const body = document.querySelector("body");

const homeAdd = function (arg) {

  arg.classList.add("home-body");
```

```javascript
}
//axios requests that reach our server to run SA on said story and show result inside button
const saButtons = document.querySelectorAll(".sa-button");

  for (let button of saButtons) {

                                          button.addEventListener("click", async function (evt) {
                                            evt.preventDefault();
                                            const text = button.firstChild;
                                            const input = text.nextSibling;
                                            const value = input.value;
                                            const storyID = button.getAttribute('data-story')
                                            if (value === "Get Polarity") {
                                              try {
                                                const req = await axios.post(`/story/${storyID}/polarity`);
                                                const resp = req.data.response;
                                                input.value = resp;
                                              } catch (error) {
                                                console.error(error.response.data)
                                              }
                                            }
                                            else if (value === "Get Subjectivity") {
                                              const req = await axios.post(`/story/${storyID}/subjectivity`);
                                              const resp = req.data.response;
                                              input.value = resp;
                                            }
                                          })
                                        }

                                        const deleteQueryButtons = document.querySelectorAll(".close")
                                        for (let button of deleteQueryButtons) {
                                          button.addEventListener("click", async function (evt) {
                                            console.log("CLICKEDDDD")
                                            const queryID = button.getAttribute('data-query')
                                            console.log(queryID)
                                            const req = await axios.post(`/user/${queryID}/delete`);
                                            const resp = req.data.response;
                                            button.value = resp;
                                          })
                                        }


                                        //Replace broken images with default image
```

```javascript
const default_avatar = 'https://secure.gravatar.com/avatar?d=wavatar';

window.addEventListener("load", event => {
  let images = document.querySelectorAll('img');
  for (let image of images) {
    let isLoaded = image.complete && image.naturalHeight !== 0;
    if (!isLoaded) {
      image.src = default_avatar;
    }
  }
});
// Source: https://www.techiedelight.com/replace-broken-images-with-javascript/
```

8. ADVANTAGES & DISADVANTAGES

➤ This app can be accessed anywhere and anytime, So that the user can view the news
➤ Its ad free
➤ The news is only based on the API
➤ It may contain some unwanted content but we don't have control over it
➤ The user can bookmark their favourite news

9. CONCLUSION

Thus we have developed a full stack application based on the plans and within the given time. We have tested the application in both desktop and mobile and it worked well, Overall it was a great experience.

10. FUTURE SCOPE

In future we may integrate our own news API instead of third party APIs and may develop a mobile native application so that it can be used in both android and ios.

11. APPENDIX
https://github.com/IBM-EPBL/IBM-Project-35519-1660285545