**NEWS TRACKER APPLICATION**

**PROJECT BASED LEARNING**

## Submitted by

## DEEPIKA A N (910619104015)

## BRIGITAMAGRINABISHAG R (910619104011)

## CHITRA DEVI G (910619104012)

## DIVYA BHARATHI G G (910619104018)

## DURGA SHEELA R (910619104019)

in partial fulfillment for the award of the degree

## of

**BACHELOR OF ENGINEERING**

## in

**COMPUTER SCIENCE AND ENGINEERING**

## K.L.N. COLLEGE OF ENGINEERING, POTTAPALAYAM ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2020

## ANNA UNIVERSITY : CHENNAI 600 025 BONAFIDE CERTIFICATE

Certified that this project report **“NEWS TRACKER APPLICATION ”** is the bonafide work of **“DEEPIKA A N (910616104015)”,“BRIGITA MAGRIN ABISHAG R (910616104011)” ,“CHITRA DEVI G (910616104012)”,**

**“DIVYA BHARATHI G G (910619104018)”** and **“DURGA SHEELA R”(910619104019)** who carried out the project under my supervision.

**SIGNATURE**

Dr.S.MIRUNA JOE AMALI

**HEAD OF THE DEPARTMENT**

Computer science and engineering,

K.L.N. College of Engineering, Pottapalayam,

Sivagangai-630 612.

**SIGNATURE**

Mr.D.Prabhu

**ASSISTANT PROFESSOR**

Computer science and engineering,

K.L.N. College of Engineering, Pottapalayam,

Sivagangai-630 612.

Submitted for the project viva-voce conducted on 22-09-2020

MENTOR EVALUATOR

## 2TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE**  **NO.** |
|  | **ABSTRACT** | **i** |
|  | **LIST OF FIGURES** | **ii** |
|  | **LIST OF ABBREVIATIONS** | **iii** |
| **1.** | **INTRODUCTION** | **1** |
|  | * 1. Project Overview | **2**  **2** |
|  | 1.2 Purpose |  |
| **2.** | **LITERATURE SURVEY** | **3** |
|  | 2.1 Existing problem | **4** |
|  | 2.2 References | **5** |
|  | 2.3 Problem statement definition | **6** |
| **3.** | **IDEATION & PROPOSED SOLUTION** | **7** |
|  | 3.1 Empathy Map Canvas | **8** |
|  | 3.2 Ideation & Brainstorming | **10** |
|  | 3.3 Proposed Solution | **13** |
|  | 3.4 Problem Solution fit | **15** |
| **4.** | **REQUIREMENT ANALYSIS** | **16** |
|  | 4.1 Functional Requirement | **17** |
|  | 4.2 Non-Functional Requirements | **17** |
| **5.** | **PROJECT DESIGN** | **18** |
|  | 5.1 Data Flow Diagrams | **19** |
|  | 5.2 solution & Technical Architecture | **20** |
|  | 5.3 User Stories | **21** |
| **6.** | **PROJECT PLANNING & SCHEDULING 22** | |
| 33 | 6.1 sprint Planning &Estimation | **24** |
|  | 6.2 Sprint Delivery Schedule | **27** |
|  | 6.3 Reports From Jira | **28** |
| **7.** | **CODING & SOLUTIONING** | **30** |
|  | 7.1 Feature 1 | **31** |
|  | 7.2 Feature 2 | **34** |
|  | 7.3 Database schema | **36** |
| **8.** | **TESTING** | **38** |

|  |  |
| --- | --- |
|  | 8.1 Test Cases **39** |
|  | 8.2 User Acceptance Testing **39** |
| **9.** | **RESULTS 40** |
|  | 9.1 Performance Metrics **41** |
| **10.** | **ADVANTAGES AND DISADVANTAGES 44** |
| **11.** | **CONCLUSION 46** |
| **12.** | **FUTURE SCOPE 47** |
| **13.** | **APPENDIX 48** |

## ABSTRACT

In today's fast-paced and dynamic world, staying on top of the latest news and events is essential for many people. Whether to follow the developments of a global crisis, the results of a sports game, the reviews of a new movie, or the opinions of favorite influencers, it need a reliable and convenient way to access the news that matters to the user. That's why a news tracker app is a perfect solution for news consumption needs. A news tracker app is a tool that is used to browse, read, and share news stories on your mobile device. The user can choose from different sources and topics that interest the user, such as politics, sports, entertainment, and more. The user can also get personalized suggestions, alerts, and updates based on your preferences and habits. A news tracker app helps you stay updated, engaged, and informed with the world. A news tracker app is a mobile software that enables users to access, read, and share news stories from various sources and topics, with features such as personalization, notifications, online access. A news tracker app works by fetching news stories from different sources and displaying them on the device. The user can select the sources and topics that you want to follow, such as newspapers, magazines, websites, blogs, podcasts, and more. The user can also search for specific keywords, categories, or locations that you are interested in. The app will learn from your choices and behavior and recommend news stories that match the user interests and preference. A news tracker app works by providing a convenient, customized, and interactive way of accessing, reading, and sharing news stories on the device.

i

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 1.1 | **Empathy Map Canvas** | 8 |
| 1.2 | **Ideation & Brainstorming** | 9 |
| 1.3 | **Team Gathering, Collaboration and Select the Problem Statement** | 10 |
| 1.4 | **Brainstorm, Idea Listing and Grouping** | 11 |
| 1.5 | **Idea Prioritization** | 12 |
| 1.6 | **Problem Solution fit**  **Focus on J&P, tap into BE, understand RC**  **Explore AS, differentiate** | 15 |
| 1.7 | **Data Flow Diagrams** | 19 |
| 1.8 | **Solution & Technical Architecture** | 20 |
| 2.1 | **Sprint Delivery Schedule** | 27 |
| 2.2 | **Reports from JIRA** | 28 |
| 2.3 | **Sprint 1 and Sprint 2** | 29 |
| 2.4 | **Sprint 3 and Sprint 4** | 30 |
| 3.1 | **Performance Metrics** | 41 |
| 3.2 | **Login Page** | 42 |
| 3.3 | **News page** | 43 |

ii

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| **HTML** | **HYPER TEXT MARKUP LANGUAGE** |
| **CSS** | **CASCADING STYLE SHEET** |
| **SQL** | **STRUCTURED QUERY LANGUAGE** |

iii

1. **INTRODUCTION**

1

**CHAPTER 1**

**INTRODUCTION**

**1.1 Project Overview**

A news tracker app is a project that aims to develop a mobile software application that allows users to access, read, and share news stories from various sources and topics. The project will involve designing, developing, testing, and deploying the app on different platforms, such as Android, iOS, and Windows. The project will also involve creating a user interface, a database, a web service, and a news aggregator. The project will use various technologies, such as HTML, CSS, JavaScript, SQL, python flask and IBM cloud. The project will follow the agile methodology, with sprints, scrums, and user stories. The project will have a team of developers, designers, testers, and project leader. The project will have a timeline, and a scope. The project will have a goal, a vision, and a mission. The project will have a problem statement, a solution statement, and a value proposition. The project will have a target market, a user persona, and a user journey. The project will have a prototype, a minimum viable product, and a final product. The project will have a user testing, a feedback, and a evaluation. The project will have a launch, a marketing, and a maintenance. A news tracker app is a project that aims to create a mobile software application that enables users to access, read, from various sources and topics, with features such as personalization, notifications, online access, and social media integration.

* 1. **Purpose**

The purpose of the news tracker app is to provide users with a convenient, customized, and interactive way of accessing, reading on their device. The app aims to help users stay updated, engaged, and informed with the world. The app also aims to offer users a variety of sources and topics that suit their interests and preferences. The app also aims to enhance users' experience and satisfaction with the news content and the app features. The purpose of the news tracker app is to create a mobile software application that delivers news stories to users in a user-friendly, personalized, and social manner.

2

# LITERATURE SURVEY

# 3

* 1. **Existing problem**

**CHAPTER 2 LITERATURE SURVEY**

News organizations whose mobile apps only provide users with their articles or videos are missing a big opportunity. An application, by definition, should be applied to perform a task or to solve a problem. Most news doesn’t do that.

Rather than just feed readers recent stories you wrote about their problems, apps can provide tools and data that enable users to solve their problems. When you solve problems, you get more loyal users and a chance to make more money. This solves the problem of the user having to reach out to other sources or to the Internet to verify/double-check if the news was real or fake administers extra efforts on the user’s part and also demolishes the main intent of the news app to provide a single-stop credible news platform and also causes irritation to the user.

The “like metric” solves the problem of the user not understanding whether it is worth spending his energy and time on this article. Also, it helps the user indicates that it is credible information.

4

* 1. **References**

# [1] Research on Topic Detection and Tracking for Online News Texts 30 April 2019.

# [2] L. Qiu and J. Yu, "CLDA: An effective topic model for mining user interest preference under big data background", Complexity, vol. 2018, May 2018.

# [3] W.- Y. Bai, C. Zhang, K.-F. Xu and Z.-M. Zhang, "A self-adaptive microblog topic tracking method by user relationship", Acta Electronica Sinica, vol. 45, no. 6, pp. 1375-1381, Jun. 2017

# [4] M. Hasan, M. A. Orgun and R. Schwitter, "A survey on real-time event detection from the twitter data stream", Journal of Information Science, pp. 0165551517698564, 2019.

# [5] An Approach to News Event Detection and Tracking Based on Stream of Online News ,21 September 2017

5

* 1. **Problem Statement Definition**

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 every day and also understand explicitly that the news is not fake but from proper sources.

As a user

* Spamming of messages usually leads to clearing of the content without viewing thus probably leading the user to lose access to important information.
* Too many notifications irritate the users and it clutters the bar on their phone
* Irrelevant news makes the user stop viewing the news thus losing access to credible news
* Users don’t want to spend time reading the entire content. They need short and crisp news
* Using multiple apps because users are not getting the content categories they want in one app.
* The user doesn’t want to waste time figuring out the relevance of the news
* Young people don’t read news from apps, usually depend on social media to get updated
* Users sometimes get overwhelmed by too many categories
* Users only check their phones and go through these notifications when they are free or during a particular time of the day when they are doing time pass
* Ads in the apps might irritate the user while reading the news
* One app not accommodating both the regional and international news might create a requirement for different apps

As a developer, I tried to **filter out**whatever isn’t important;

* The app should ask the user initially, what categories are they interested to read from. And, show 8 out of every 10 news related to that category only (assuming that users get overwhelmed by too much info)
* **Giving filters for notifications (in terms of content category and also frequency) because assuming that users get irritated by too many filters**
* An App that includes all international and regional news that can be customizable depending on the users’ needs, will reduce the number of apps (assuming that people use more than 1 app)
* **The app should provide info about all the trusted worldwide sources, and then in each article, it should mention which source has validated this news, as I’m assuming that users can’t differentiate between real and fake news**
* An app allowing the user to choose/customize the time for notification popup for news (assuming that people only check news notifications during free time)
* Users don’t want to spend time reading the entire content. They need short and crisp news

# IDEATION & PROPOSED SOLUTION

7

## CHAPTER 3

**IDEATION & PROPOSED SOLUTION**

* 1. **Empathy Map Canvas**.

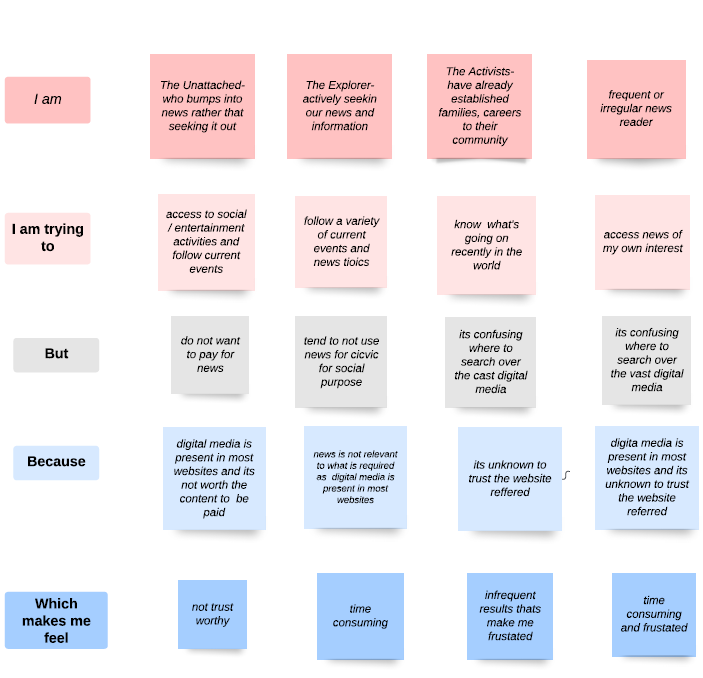
1

**s**

**Figure no 1.1**

8

* 1. **Ideation & Brainstorming**

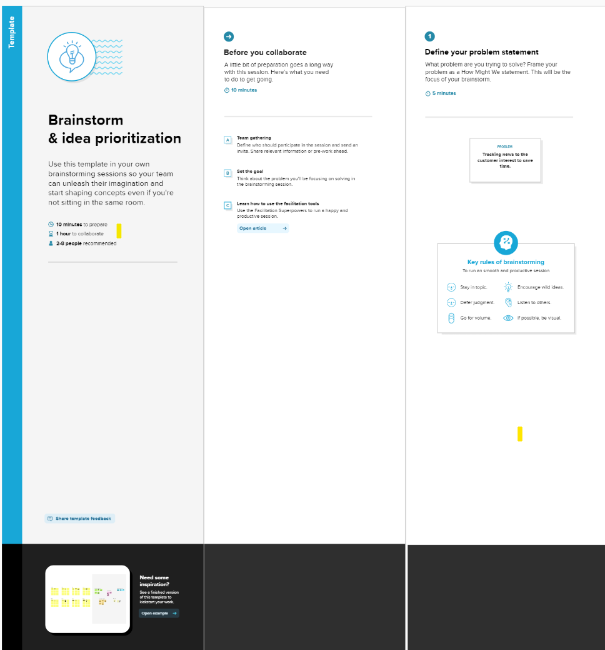


**Figure no 1.2**

9

**Brainstorm & Idea Prioritization:**

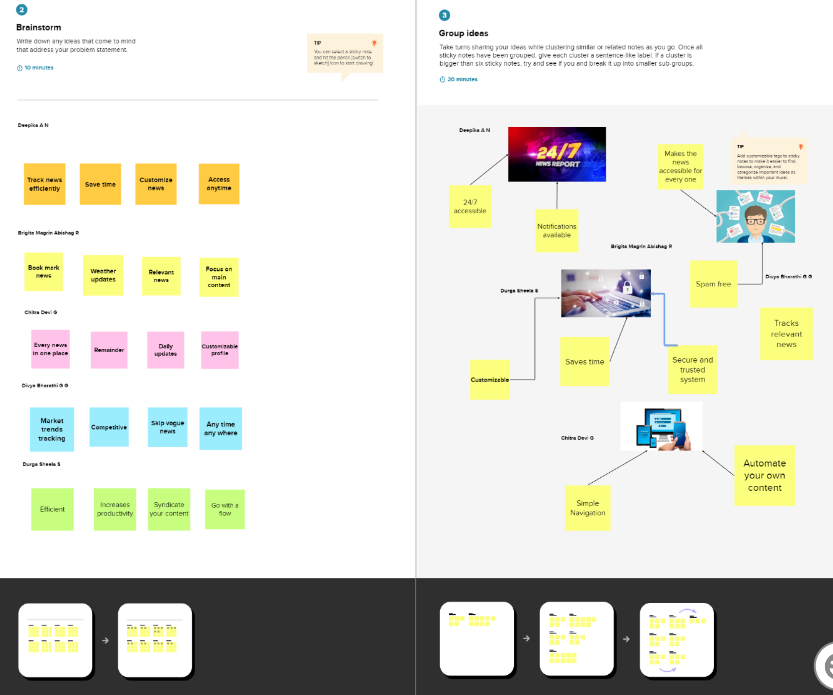
**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

****

**Figure no 1.3**

10

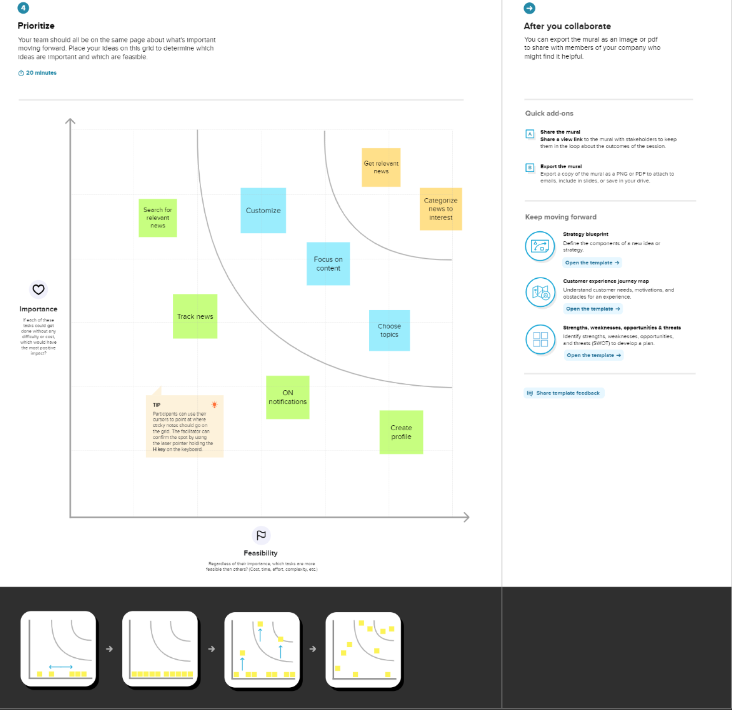
**Step-2: Brainstorm, Idea Listing and Grouping**



**Figure no 1.4**

11

**Step-3: Idea Prioritization**



**Figure no 1.5**

12

* 1. **Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **PARAMETER** | **DESCRIPTION** |
| 1. | Problem statement | 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 every day and also understand  explicitly that the news is not fake but from proper sources. |
|  | Idea / Solution description | This is where you need to focus on bringing interactive, visual and architectural designs as well. It means that the content should be distributed in the app such that the screen do not appear crowded with the  content. |

13

|  |  |  |
| --- | --- | --- |
| 3 | Uniqueness | The main focus of this application is to content news articles from all around the world and deliver it to user as fast as possible in best visualize way.  The tracker app does not have annoying ads. With less amount of usuage of data. |
| 4 | Customer  satisfaction | bug free, easy to use on both the management and user  end, and simple to understand. |
| 5 | Business model | If users decide on a medium to lengthy-term subscription, you may provide them with an ad-free  premium version of the app as well as other perks like yearly or quarterly discounts. |
| 6 | Scalability of the Solution | Since the web application is deployed on IBM cloud, it can handle multiple users at a time. The user will go through a seamless experience and it enables them to view the news according to their interests and choices.  Users from all age category can use the application and the news can also be filtered according to their age |

**Focus on J&P, tap into BE, understand RC**

**Explore AS, differentiate**

14

**3.4 Problem Solution fit**

**Focus on J&P, tap into BE, understand RC**

**Explore AS, differentiate**

**Identify strong TR & EM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identify strong TR & EM** | **3. TRIGGERS TR**  Negative moments while errors in the operation  And downloading the news feed | **10.YOUR SOLUTION:**  A web application that tracks news according to user’s interest and helps the user to keep track of day to day important news. | 1. **CHANNELS of BEHAVIOUR CH**     1. **ONLINE**   Gets up to date news on time.   * 1. **OFFLINE**   Download and save information if necessary. |  |
| **4. EMOTIONS: BEFORE / AFTER EM**  Before: Need manual access to information Lack of awareness.  After: Get news at anywhere and anytime Being aware of current affairs  . |

i.e. directly related: ﬁnd the right solar panel installer, calculate usage and beneﬁts; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

**BE**

**7. BEHAVIOUR**

Get current updates on featured news

Get notification alerts

Can view news offline also

**RC**

**9. PROBLEM ROOT CAUSE**

.

Difficulty in accessing the news for offline users

Unable to get news on time.

**J&P**

**2. JOBS-TO-BE-DONE / PROBLEMS**

For Latest news updates

Breaking news

Tracking news according to user’s interest

Save user’s time

**Focus on J&P, tap into BE, understand RC**

**Focus on J&P, tap into BE, understand RC**

.

Newspaper were the exsisting system.

Live updates of news in television.

**AS**

**5. AVAILABLE SOLUTIONS**

Available devices

Network connection

**CC**

**6. CUSTOMER CONSTRAINTS**

**CS**

**1. CUSTOMER SEGMENT(S)**

Users

**Explore AS, differentiate**

**Define CS, fit into CC**

15

**Figure no 1.6**

15

# 

# REQUIREMENT ANALYSIS

# 

# 

# 16

## 

## CHAPTER 4 REQUIREMENT ANALYSIS

* 1. **Functional requirement**

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

* 1. **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 |

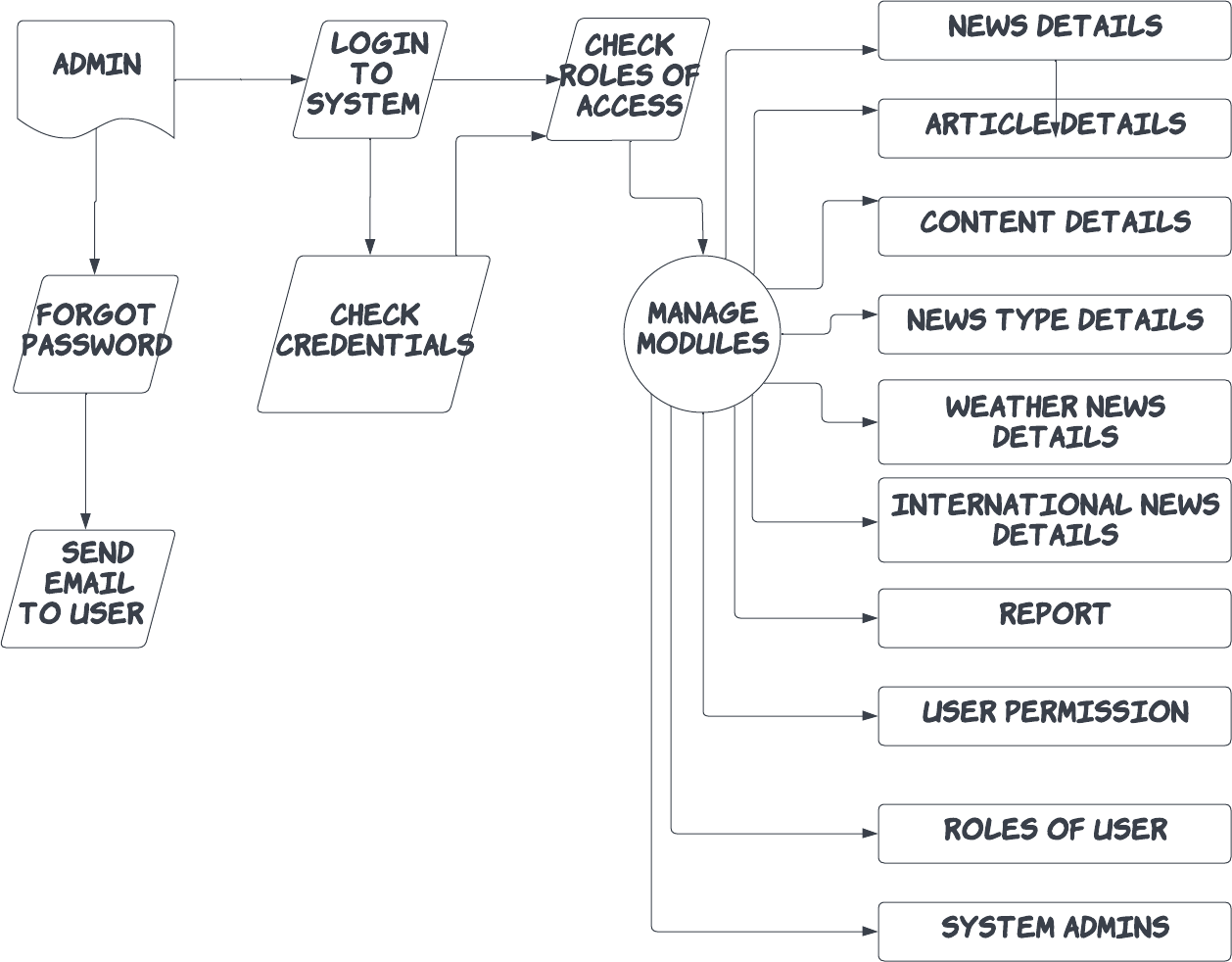
17

# PROJECT DESIGN

18

* 1. **Data Flow Diagrams**

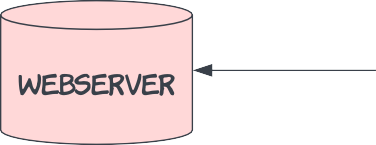
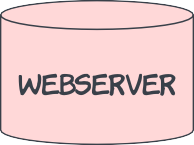
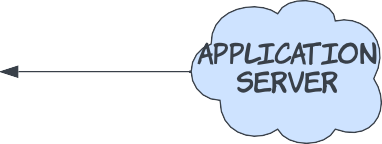
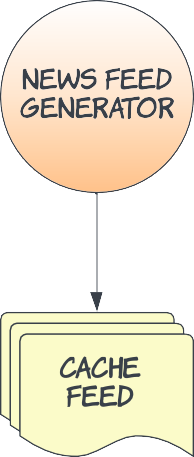
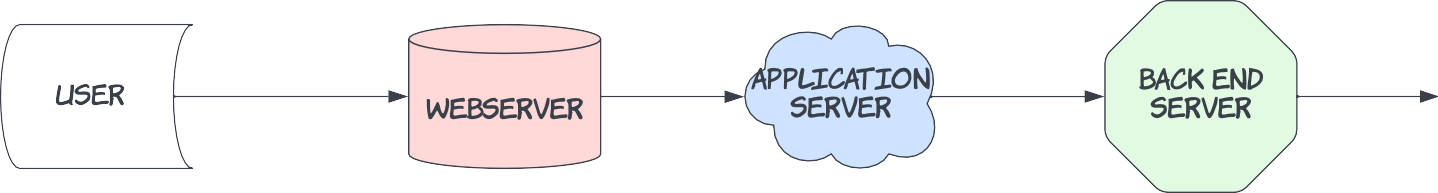
**CHAPTER 5 PROJECT DESIGN**



**Figure no 1.7**

19

* 1. **Solution & Technical Architecture**



**Figure no 1.8**

20

* 1. **User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement**  **(Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (mobile user) | Installation | USN-1 | As a user, I  installed the app  for day-to-day update and  feeds. | My app  will be  installed on  home  screen |  |  |
| Customer  (Mobile  user) |  | USN-2 | As a user, I can  register for the  application by  entering my  email, password,  confirming  my  password  and phone  number. | I can access  my account /  dashboard | High | Sprint-1 |
|  |  | USN-3 | As a user, I  will receive  conformation  email once I  have  registered for  the | I can receive  conformation  email & | High | Sprint-1 |
| USN-4 | As a user, I can register for the application  Through Gmail | I can register  & access the dashboard  with  Gmail account  Login | Medium | Sprint-1 | USN-4 | Sprint-1 |

21

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Login | USN-5 | As a user, I can  log into the  application by  entering email &  password | I can login  to the  official  page | High | Sprint-1 |
|  | Dash board | USN-6 | Day to day news, feeds, categories, tech news and other updates | I can see all the news which I wanted | High | Sprint-1 |
| Customer  (Web  user) | Browsing | USN-7 | Enter the website on the browser | I can even  login  through | Medium | Sprint-1 |

22

# PROJECT PLANNING & SCHEDULING

23

## CHAPTER 6

**PROJECT PLANNING & SCHEDULING**

* 1. **Sprint Planning & Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional requirement(epic)** | **User story number** | **User story/task** | **Story points** | **Priority** | **Team members** |
| Sprint- 1 | Setting up application | USN -1 | As a user, I can register in ICTA academy and I can create an IBM account. | 3 | high | Deepika A N |
| Sprint- 1 |  | USN -2 | As a user, I can create a flask project | 3 | high | Brigita Magrin Abishag R |
| Sprint- 2 | Setting up Application | USN -3 | As a user, I can create a flask project. | 2 | high | Chitra Devi G |
| Sprint- 2 |  | USN -4 | As a user, I will create a login page  And pass word | 2 | high | Divya Bharathi G G |

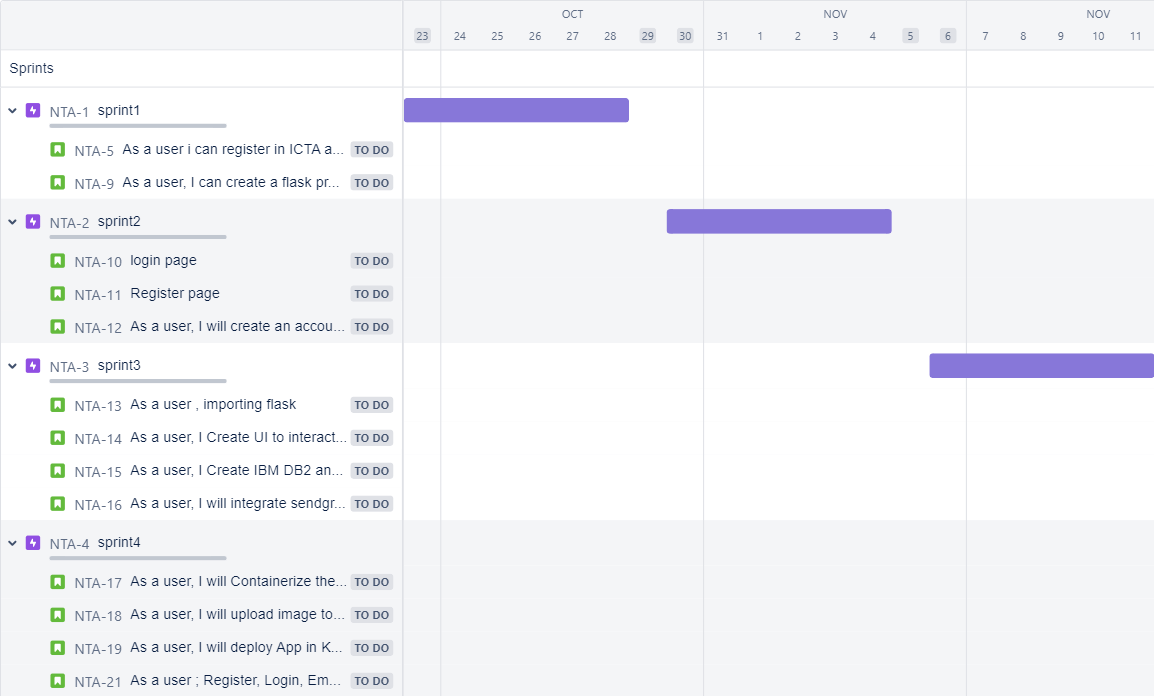
24

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint- 2 |  | USN -5 | As a user, I will create a register page | 1 | low | Durga Sheela S |
| Sprint- 2 |  | USN -6 | As a user, I will create an account in send grid | 1 | low | Deepika A N |
| Sprint-  3 | Implementing Web Application | USN -7 | As a user importing flask | 1 | low | Brigita Magrin Abishag R |
| Sprint-  3 |  | USN -8 | As a user, I create UI to interact with the application | 2 | medium | Chitra Devi G |
| Sprint-  3 |  | USN -9 | As a user, I create IBM DB2 and connect with the python | 1 | low | Divya Bharathi G G |
| Sprint- 3 | Integrating Send Grid Service | USN - 10 | As a user, I will integrate send grid with the python | 2 | medium | Durga Sheela S |
| Sprint- 4 | Development of App in IBM cloud | USN - 11 | As a user, I will containerize is the IBM cloud | 1 | low | Deepika A N |
| Sprint- 4 |  | USN - 12 | As a user, I will upload the image to the IBM container registry | 2 | High | Brigita Magrin Abishag R |
| Sprint- 4 |  | USN - 13 | As a user, I will deploy the app in the Kubernetes cluster | 2 | High | Durga Sheela S |
| Sprint 4 | User Panel | USN -14 | As a user;  Register, log in, email verification, and manually search | 1 | low | Chitra Devi G |

26

* 1. **Sprint Delivery Schedule**

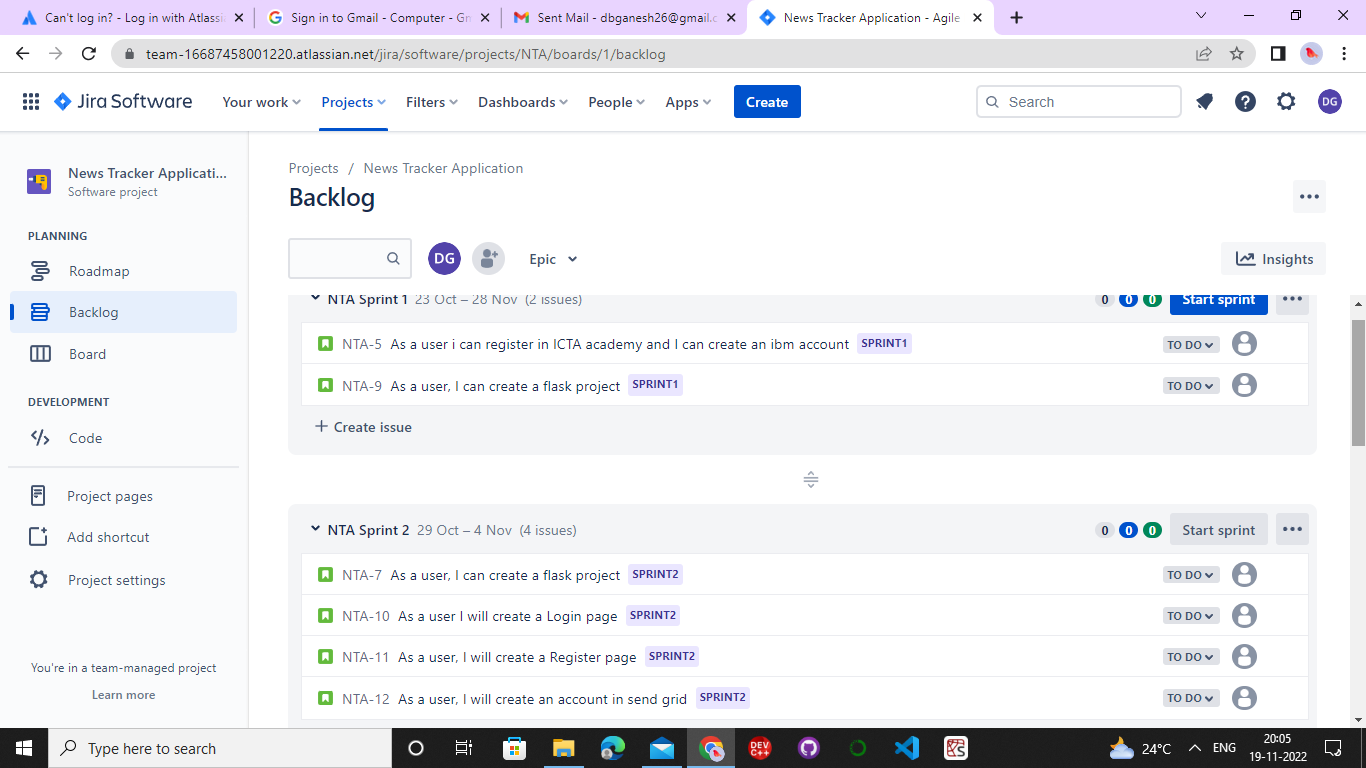
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total story points** | **Duration** | **Sprint Start Date** | **Sprint End Date** (Planned) | **Story Points Completed** (as on Planned End Date) | **Sprint Release Date** (Actual) |
| Sprint 1 | 18 | 6 Days | 23 Oct 2022 | 28 Oct 2022 | 24 | 28 Oct 2022 |
| Sprint 2 | 18 | 6 Days | 30 Oct 2022 | 04 Nov 2022 | 24 | 04 Nov 2022 |
| Sprint 3 | 18 | 6 Days | 06 Nov 2022 | 11 Nov 2022 | 24 | 11 Nov 2022 |
| Sprint 4 | 18 | 6 Days | 13 Nov 2022 | 18 Nov 2022 | 24 | 18 Nov 2022 |

****

**Figure no 2.1**

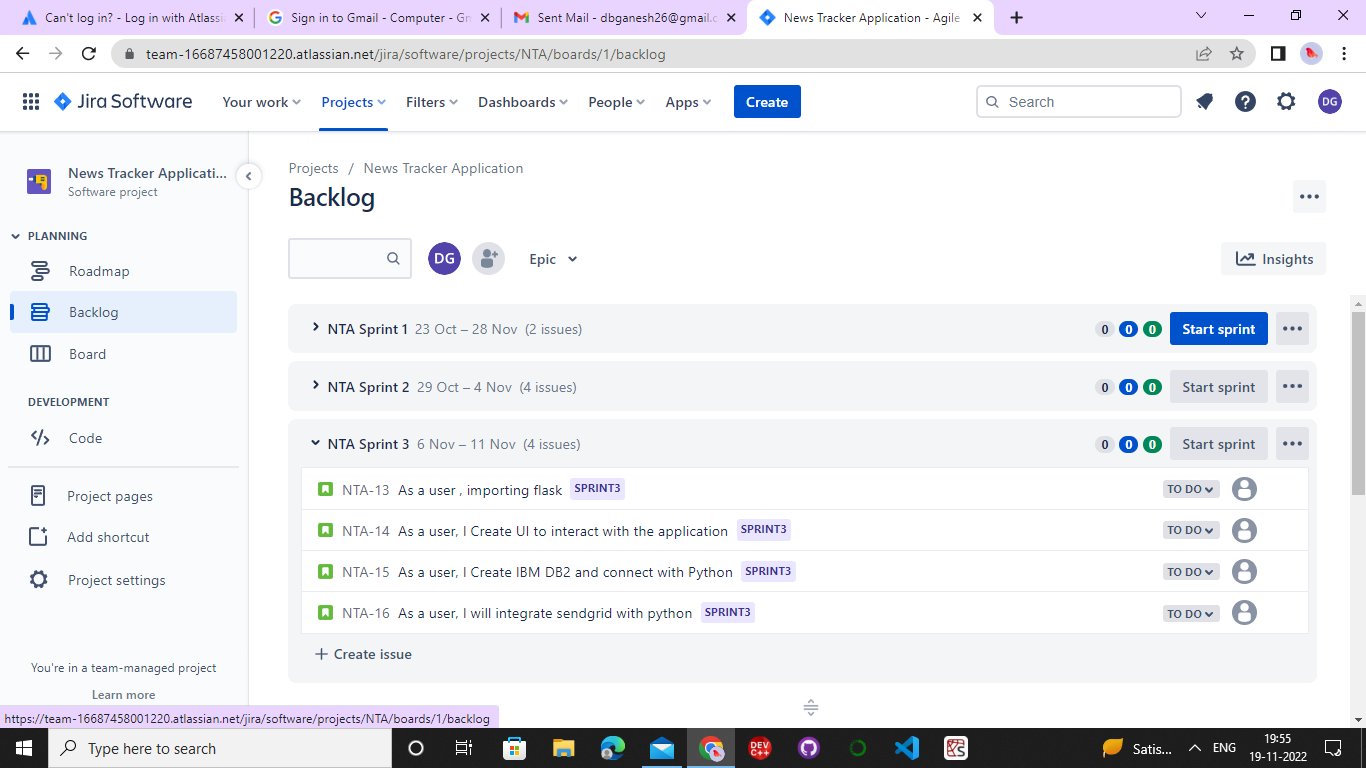
27

* 1. **Reports from JIRA**

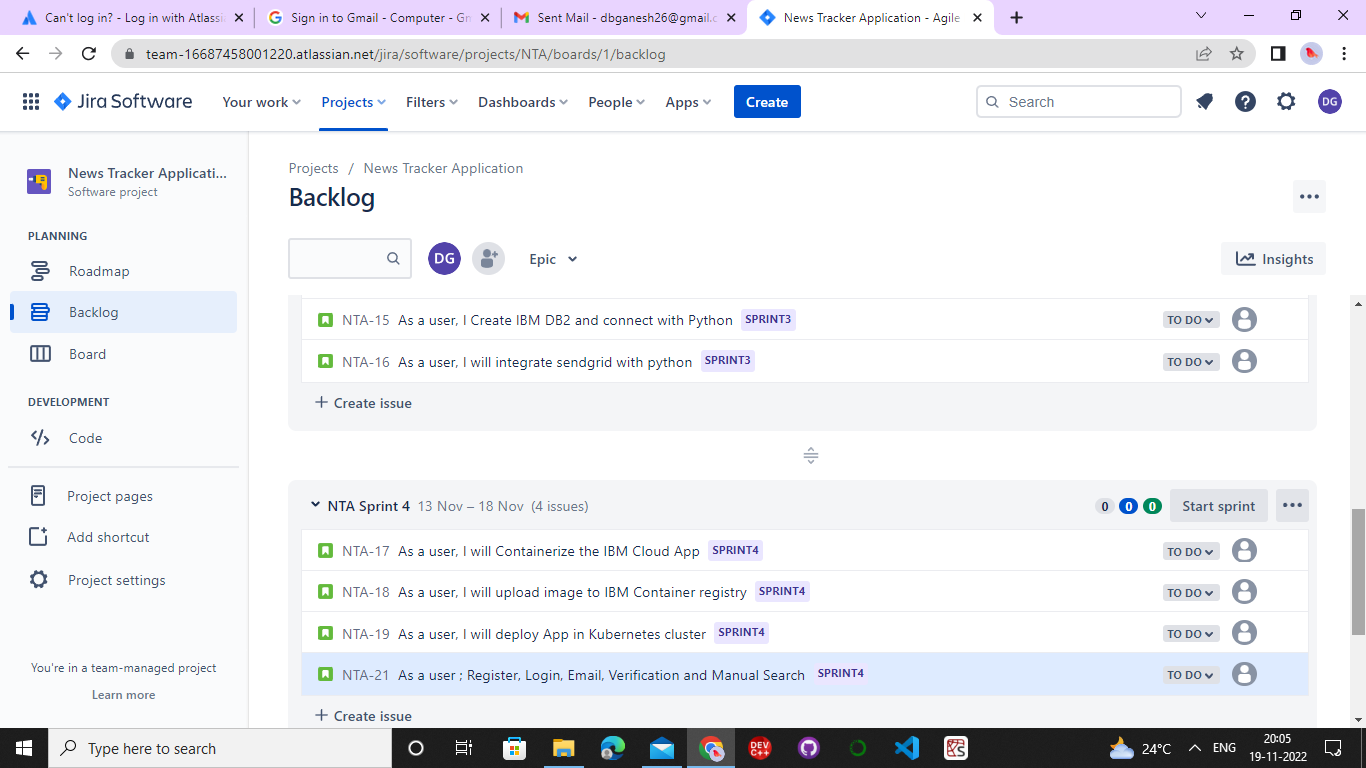
****

**Figure no 2.2**

28

****

**Figure no 2.3**



**Figure no 2.4**

29

# CODING & SOLUTIONING

# 30

**7.1 Feature 1**

**Login.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">

<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/uc?id=16U\_\_U5dJdaTfNGobB\_OpwAJ73vM50rPV&export=download"

alt="">Return home</a></div>

<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">EXPLORE</button>

</form>

</div>

</div>

<div class="right">

<div class="right-text">

<h2>News Application</h2>

<h5>Gain knowledge ,Save time</h5>

</div>

</div>

</section>

</body>

</html>

31

**Login.css**

img {

width: 100%;

}

.login {

height: 1000px;

width: 100%;

background: radial-gradient(#6d587e, #1b1a1c);

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: #7f6dab;

font-weight: 400;

}

.left .top\_link {

height: 20px

}

.left .contact {

display: flex;

align-items: center;

justify-content: center;

align-self: center;

height: 100%;

width: 73%;

margin: auto;

}

.left h3 {

32

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);

}

.right {

background: linear-gradient(212.38deg, rgba(183, 55, 27, 0.7) 0%, rgba(175, 70, 189, 0.71) 100%), url(https://www.pewresearch.org/wp-content/uploads/sites/8/2016/07/PJ\_2016.07.07\_Modern-News-Consumer\_0-01.png) no-repeat;

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%;

33

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;

}

**7.2 Feature 2**

**Index.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>SignUp</title>

<link rel="Stylesheet" href="index.css">

</head>

<body>

<nav>

<ul class="navigation">

<div class="brand"><img src="#" alt="LOGO" />

<!-- Making menu icon clickable to display the navigation menu on smaller screens -->

<i onclick="navToggle()" id="nav-icon" class="fa fa-navicon" style="font-size:24px"></i>

</div>

<div id="toggle" class="nav-container">

<a class="left" href="news.html">

<li>GO BACK HOME</li>

</a>

</div>

</ul>

</nav>

<!-- NAVIGATION END HERE -->

<section class="form">

<div class="center">

34

<h1>GET STARTED<b style="color: #daa800 ;"> 33

WITH OUR NEWS APPLICATION</b> TO KNOW ABOUT THE WORLD NEWS</h1>

<hr width="20%" style="border: 1px solid #daa800;">

<br>

<form action="">

<input class="name-surname" type="text" name="name" placeholder="Name">

<input class="name-surname" type="text" name="surname" placeholder="Surname"><br>

<input type="text" name="email" placeholder="Email"><br>

<div class="checkbox">

<h3 style="color: #daa800 ;"><span>SELECT THE CATEGORY </span><br>

<hr>

</h3>

<input type="checkbox" name="Categories" id="p1">

<label for="p1">World News</label>

<input type="checkbox" name="Categories" id="p2">

<label for="p2">Weather News</label><br>

<input type="checkbox" name="Categories" id="p3">

<label for="p3">Sports News </label>

<input type="checkbox" name="Categories" id="p4">

<label for="p4">Technology</label><br>

<input type="checkbox" name="Categories" id="p5">

<label for="p5">Political News</label>

<input type="checkbox" name="Categories" id="p6">

<label for="p6">Trending Today</label> <br><br>

</div>

<br> <input type="password" name="password" placeholder="Password"><br>

<input type="password" name="conf\_password" placeholder="Confirm Password"><br>

<button formaction="news.html">JOIN</button>

<p>Already have an account? <a href="login.html">Login Here</a></p>

</form>

</div>

</section>

<!-- FOOTER STARTS HERE -->

<footer>

<div class="footer\_container">

<p class="bottom\_text">Company Name - Copyright 2018 All Rights Reserved.</p>

35

</div>

</footer>

<!-- FOOTER ENDS HERE -->

</body>

</html>

**DATA BASE SCHEMA:**

**App.py**

from glob import escape

from flask import Flask, render\_template, request, redirect, url\_for, session

import ibm\_db

app = Flask(\_\_name\_\_)

#conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pss83307;PWD=Lab47l4rodOvB2WE",'','')

@app.route('/')

def login():

return render\_template('signin.html')

@app.route('/signin.html',methods = ['POST'])

def getUser():

if request.method == 'POST':

user = request.form['uname']

password = request.form['upwd']

sql = "SELECT \* FROM CUSTOMERS where Email = ?"

stmt = ibm\_db.prepare(conn, sql)

email = user

# Explicitly bind parameters

ibm\_db.bind\_param(stmt, 1,user)

ibm\_db.execute(stmt)

dictionary = ibm\_db.fetch\_assoc(stmt)

pwd = dictionary["PASSWORD"]

if password != pwd:

return render\_template('error.html')

return render\_template('base.html')

@app.route('/signup.html')

def putUser():

return render\_template('signup.html')

@app.route('/signup.html',methods = ['POST'])

def storedUser():

36

if request.method == 'POST':

fname = request.form['fname']

lname = request.form['lname']

mail = request.form['mail']

npwd = request.form['npwd']

cpwd = request.form['cpw’}

res = fname + lname + mail + npwd + cpwd

if npwd != cpwd:

return render\_template('signup.html')

sql = "INSERT INTO customers (FirstName,LastName,Email,password,confirmpassword) VALUES(?,?,?,?,?);"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt, 1, fname)

ibm\_db.bind\_param(stmt, 2, lname)

ibm\_db.bind\_param(stmt, 3, mail)

ibm\_db.bind\_param(stmt, 4, npwd)

ibm\_db.bind\_param(stmt, 5, cpwd)

ibm\_db.execute(stmt)

return render\_template('signin.html')

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

# run() method of Flask class runs the application

# on the local development server.

app.run(debug=True)

37

**8.TESTING**

38

**CHAPTER 8**

**TESTING**

**8.1 Test Cases**

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

**8.2 User Acceptance Testing**

Purpose of Document: The purpose of this document is to briefly explain the test coverage and open issues of the News Tracker Application project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis: This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

39

# RESULTS

# 

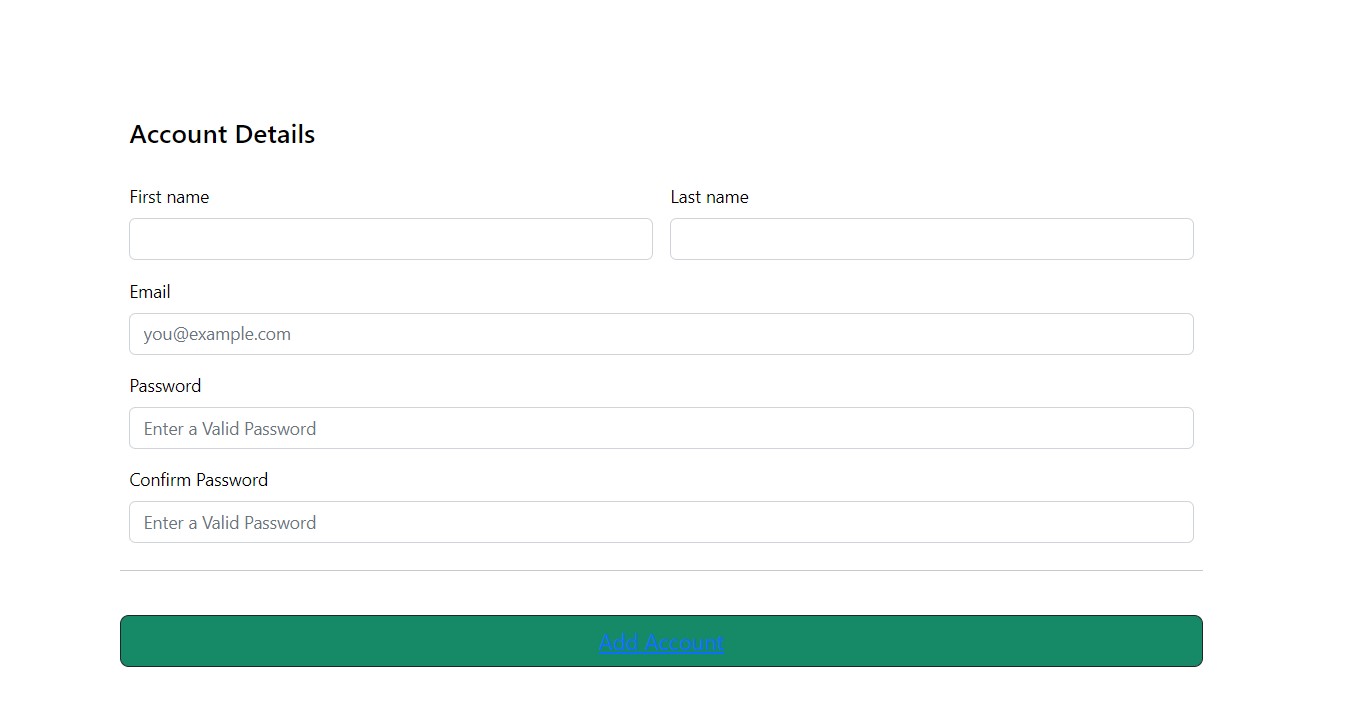
# 40

**CHAPTER 9**

**RESULTS**

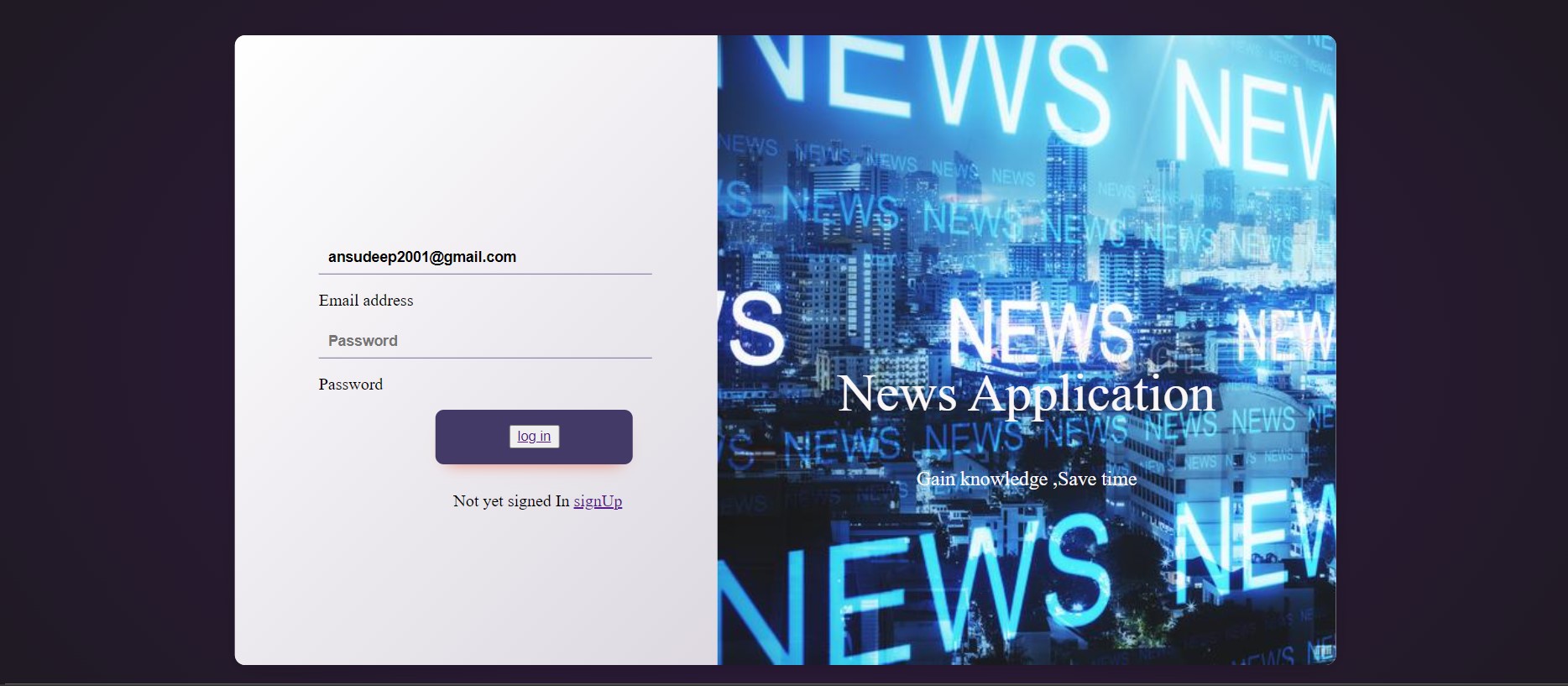
**9.1 Performance Metrics**

These application performance metrics, commonly known as key performance indicators (KPIs), are a quantitative measure of how effectively the organization achieves the business objectives. Capturing the right metrics will give you a comprehensive report and powerful insights into ways to improve your application.



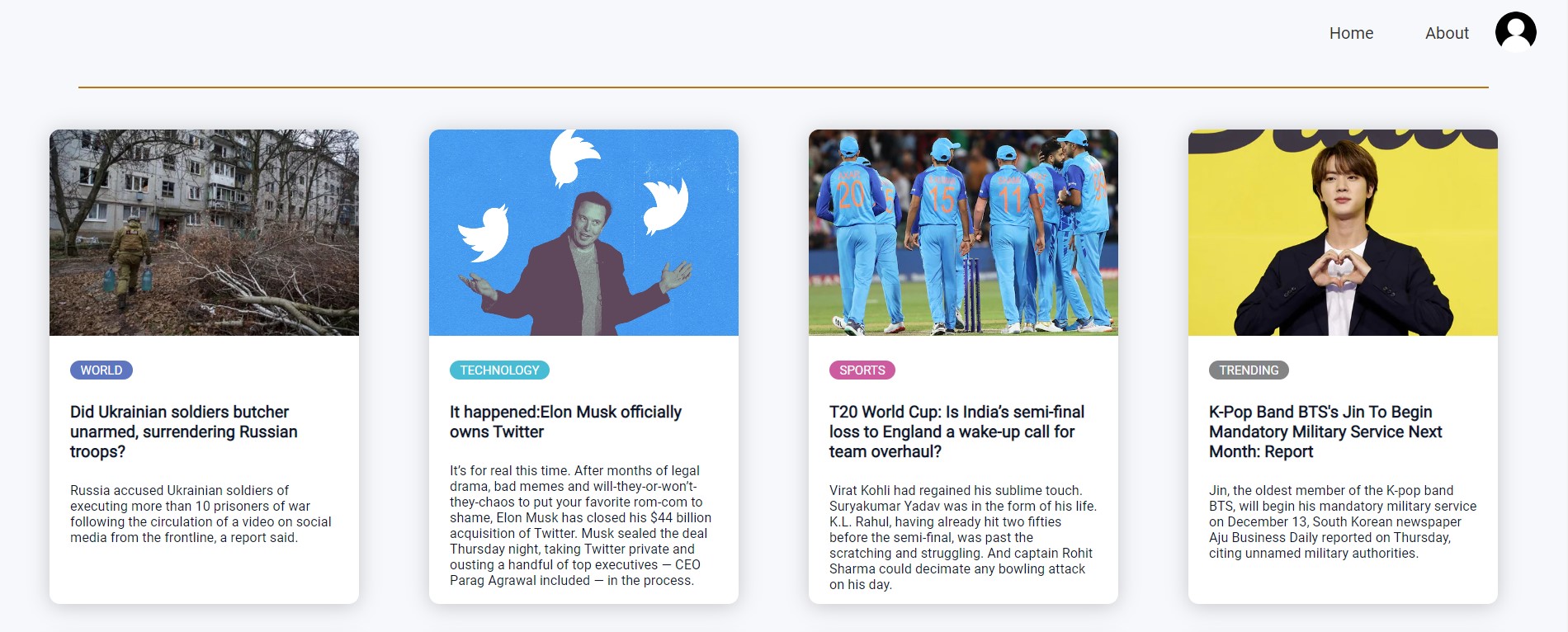
**Figure no 3.1**

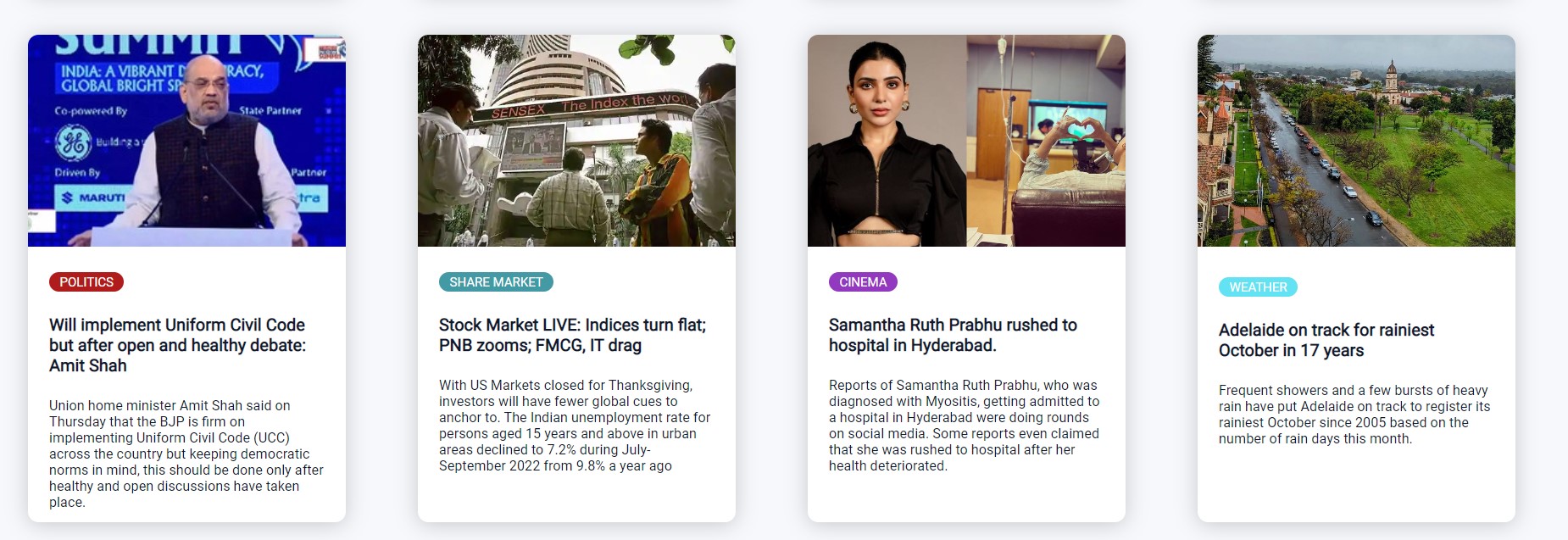
41



**Figure no 3.2**

42





**Figure no 3.3**

43

# ADVANTAGES & DISADVANTAGES

44

## CHAPTER 10 ADVANTAGES & DISADVANTAGES

**ADVANTAGE**

* Real-time application is allowed and has live layout.
* It has a Built-in support for Google Platform.
* News accuracy of specialized API Authenticated users are allowed
* Up to date and daily information is available.

## Disadvantages

* + Require data/Wi-fi to get online.
  + Companies not making as much money due to free reading for audiences.
  + News spreads quicker online - people find out news before they should.
  + Lose money - can't get people to pay for digital.
  + Older audiences may not access digital platforms.

45

# CONCLUSION

# 46

**11. CONCLUSION**

In our project work, an attempt has been made to develop a News or information-based website. We develop this project that helps people and make them aware so that they can know any news. To establish this website we use various methodologies. To develop this project we faced many problems but we hardly tried to develop this project. The project we've offered here is just the beginning of a new way of interacting with our society. In the meantime, don't forget that compelling visual content will help you be more visible and viral than offline or online newspapers.

**12. Future Scope**

* + We are in a process of developing a algorithm that will help the user to read the new postings and news from his recent data sources.
  + In pandemic situations, offline news won't be delivered to anyone, in those time these news apps are the most suitable.
  + In the future, we are going to develop a new categories according to their user locations.

47

**13.APPENDIX**

**Source Code**

import json

import bcrypt

import ibm\_db

import requests

from flask import (Flask, redirect, render\_template, request)

app = Flask(\_name\_)

# ============================================ for database with IBM===========================

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SSLServerCertificate=credientials/DigiCertGlobalRootCA.crt;UID=bgh36879;PWD=TvSzQQQ6Jhaaiwg6",'','')

print(conn)

print("🟢 connection successfull with IBM\_DB ⚡")

# signup form data

@app.route('/')

def index():

return render\_template('signup.html')

# signup form validation

@app.route('/signUpFormData',methods = ['POST', 'GET'])

def signUpFormData():

if request.method == "POST":

userName = request.form.get("userName",False)

userEmail = request.form.get("userEmail")

userPassword = request.form.get("userPassword")

userConfirmPassword = request.form.get("userPasswordConfirm")

userMobile = request.form.get("userMobile")

picture = request.form.get("picture")

if userPassword == userConfirmPassword:

sql = "SELECT \* FROM news\_tracker\_application WHERE userEmail =?"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt,1,userEmail)

ibm\_db.execute(stmt)

account = ibm\_db.fetch\_assoc(stmt)

# print(account)

bytes = userPassword.encode('utf-8')

salt = bcrypt.gensalt()

hashed\_password = bcrypt.hashpw(bytes, salt)

48

userPassword = hashed\_password

if account:

return render\_template('login.html', msg="You are already a member, please login using your details")

else:

insert\_sql = "INSERT INTO news\_tracker\_application VALUES (?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, userName)

ibm\_db.bind\_param(prep\_stmt, 2, userEmail)

ibm\_db.bind\_param(prep\_stmt, 3, userPassword)

ibm\_db.bind\_param(prep\_stmt, 4, userMobile)

ibm\_db.bind\_param(prep\_stmt, 5, picture)

ibm\_db.execute(prep\_stmt)

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail

message = Mail(

from\_email='applicationnewstracker@gmail.com',

to\_emails=userEmail,

subject='Welcome to News Tracker Application',

html\_content='<img src="https://cloud-object-storage-18-cos-standard-yx0.s3.jp-tok.cloud-object-storage.appdomain.cloud/welcom\_nta.gif" />')

try:

sg = SendGridAPIClient('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')

response = sg.send(message)

print(response.status\_code)

print(response.body)

print(response.headers)

except Exception as e:

print(str(e))

return render\_template('login.html', msg="user Data saved successfuly.. Please login use your credentials")

else:

return render\_template('signup.html', msg = 'Password and Confirm Password are not matched' )

# ============================================= for serve ======================================

49

# login form validation

@app.route('/loginForm', methods=['GET', 'POST'])

def loginForm():

if request.method == 'POST':

global email

email = request.form['userEmail']

pwd = request.form['userPassword']

var = email

sql = "SELECT \* FROM news\_tracker\_application WHERE userEmail =?"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt, 1, email)

ibm\_db.execute(stmt)

auth\_token = ibm\_db.fetch\_assoc(stmt)

print("auth",auth\_token)

if auth\_token:

# encoding user password

userBytes = pwd.encode('utf-8')

byte\_pwd = bytes(auth\_token['USERPASSWORD'], 'utf-8')

# checking password

result = bcrypt.checkpw(userBytes, byte\_pwd)

if result:

print("succ")

url = (' https://newsapi.org/v2/top-headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

TopHeadlinesResponse = requests.get(url).json()

return render\_template('index.html', msg="Logged in Successfully", responseData=TopHeadlinesResponse, tmp = 1)

else:

return render\_template('login.html', msg="Invalid Credentials", tmp = 0)

else:

return render\_template('signup.html', msg="User doesn't exist, Please Register using your details!")

else:

return render\_template('login.html', title='Sign In')

# home page

@app.route('/home')

def userdata():

print(email)

url = (' https://newsapi.org/v2/top-headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

TopHeadlinesResponse = requests.get(url).json()

return render\_template('index.html',responseData=TopHeadlinesResponse)

50

# signup form

@app.route('/')

@app.route('/signup')

def signUp():

return render\_template('signup.html')

# login form

@app.route('/login')

def login():

return render\_template('login.html')

# logout

@app.route('/logout')

def logout():

return redirect('/login')

# redirect Home

@app.route('/redirectHome')

def redirectHome():

return redirect('/home')

# about us

@app.route('/aboutus')

def aboutus():

return render\_template('aboutus.html')

# weather

@app.route('/weather')

def weather():

return render\_template('weatherinfo/weatherpage.html')

# education

@app.route('/education')

def education():

value = 'education'

crimenews = ('https://newsapi.org/v2/everything?' 'q='+value+'&''from=2022-10-29&''sortBy=popularity&''apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

educationResponse = requests.get(crimenews).json()

print(educationResponse)

# return render\_template('NewsTemplate.html',responseData=crimeNewsresponse) dharun API key = 7c7062c3a98649b5bc6ffda7fdc5a01b aravindh = 9b6f57afe98440b8b362b1046559d71d

result\_count = educationResponse.get('totalResults')

if(result\_count>0):

return render\_template('NewsTemplate.html',responseData=educationResponse,returned\_input\_search\_value=value,result\_count=result\_count)

else:

return render\_template('notfound.html')

51

# Top headlines

@app.route('/TopHeadlines')

def TopHeadlines():

value ='Top Headlines'

url = (' https://newsapi.org/v2/top-headlines?country=in&apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

TopHeadlinesResponse = requests.get(url).json()

result\_count = TopHeadlinesResponse.get('totalResults')

return render\_template('NewsTemplate.html',responseData=TopHeadlinesResponse,returned\_input\_search\_value=value,result\_count=result\_count)

# science news

@app.route('/sciencenews')

def crimenews():

value ='science'

sciencenews = ('https://newsapi.org/v2/everything?'

'q='+value+'&'

'from=2022-10-29&'

'sortBy=popularity&'

'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

scienceNewsresponse = requests.get(sciencenews).json()

print(scienceNewsresponse)

# dharun API key = 7c7062c3a98649b5bc6ffda7fdc5a01b aravindh = 9b6f57afe98440b8b362b1046559d71d

result\_count =scienceNewsresponse.get('articles')

result\_count = len(result\_count)

if(result\_count>0):

return render\_template('NewsTemplate.html',responseData=scienceNewsresponse,returned\_input\_search\_value=value,result\_count=result\_count)

else:

return render\_template('notfound.html')

# health news

@app.route('/healthnews')

def healthnews():

value = 'health'

healthnews = ('https://newsapi.org/v2/everything?'

'q='+value+'&'

'from=2022-10-29&'

'sortBy=popularity&'

'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

healthNewsresponse = requests.get(healthnews).json()

result\_count = healthNewsresponse.get('totalResults')

if(result\_count>0):

return

52

render\_template('NewsTemplate.html',responseData=healthNewsresponse,returned\_input\_search\_value=value,result\_count=result\_count)

else:

return render\_template('notfound.html')

# sports news

@app.route('/sportsnews')

def sportsnews():

value = 'sports'

sportsnews = ('https://newsapi.org/v2/everything?'

'q='+value+'&'

'from=2022-10-29&'

'sortBy=popularity&'

'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

sportsNewsresponse = requests.get(sportsnews).json()

# return render\_template('NewsTemplate.html',responseData=crimeNewsresponse)

result\_count = sportsNewsresponse.get('totalResults')

if(result\_count>0):

return render\_template('NewsTemplate.html',responseData=sportsNewsresponse,returned\_input\_search\_value=value,result\_count=result\_count)

else:

return render\_template('notfound.html')

@app.route('/searchResults', methods =["POST"])

def searchResults():

if request.method == "POST":

search\_value\_name = request.form.get("searchvalue")

print(search\_value\_name)

searchURL = ('https://newsapi.org/v2/everything?'

'q='+search\_value\_name+'&'

'from=2022-10-29&'

'sortBy=popularity&'

'apiKey=7c7062c3a98649b5bc6ffda7fdc5a01b')

searchResponse = requests.get(searchURL).json()

result\_count = searchResponse.get('totalResults')

print(result\_count) # NUMBER

if(result\_count>0):

return render\_template('NewsTemplate.html',responseData=searchResponse,returned\_input\_search\_value=search\_value\_name,result\_count=result\_count)

else:

return

53

render\_template('notfound.html',responseData=searchResponse,returned\_input\_search\_value=search\_value\_name)

# tab user

@app.route('/tabuser')

def tabuser():

userEmail = email

print('email',userEmail)

sql = "SELECT \* FROM news\_tracker\_application WHERE userEmail =?"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt, 1, userEmail)

ibm\_db.execute(stmt)

auth\_token = ibm\_db.fetch\_assoc(stmt)

return render\_template('userinfo.html', msg=auth\_token)

# logout

@app.route('/logout')

def logoutform():

email = ''

return render\_template('login.html', msg= 'successfully logged out')

#================================= server details ======================================

if \_name=='main\_':

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

GITHUB LINK: <https://github.com/IBM-EPBL/IBM-Project-36902-1660298664>

PROJECT DEMO LINK: <https://www.youtube.com/watch?v=BKxsVUYIAgM>

55