**Project Report**

**TEAM ID :** PNT2022TMID28519

**PROJECT :** NEWS TRACKER

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
8. **ADVANTAGES & DISADVANTAGES**
9. **CONCLUSION**
10. **FUTURE SCOPE**
11. **11. APPENDIX**

Source Code

# 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

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. Sometimes may show irrelevant and updated news.

## 2.2 Survey

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl.**  **No** | **TITLE** | **AUTHOR**  **S** | **YEAR** | **TECHNIQUES** | **MERITS** | **DEMERITS** |
| **1** | **Design and**  **Implementation of News**  **Collecting and**  **Filtering System Based on RSS** | Zheng, R., & Zhang, Y. | **2012** | Using RSS to collect  News with enhanced search system | System can automatically collect the latest news information from the subscribe site, then parsing and Storing the information into database. | Graphics and photos do not always appear .Posts are easily deformatted or fully erased. RSS might make the process a bit unpleasant. |
| **2** | **News Event**  **Detection and**  **Tracking Based on Stream of**  **Online News** | Yajie Qi  Li Zhou  Huayou Si  JianWan  Ting Jin | **2017** | Single-pass clustering algorithm for event detection and tracking | Extraction of news content on particular field by searching keywords. | In some cases it can’t analyse the keyword brings original news or not |
| **3** | **Deep News**  **Event Ranker**  **Based on User**  **Relevant Query** | Kong, X.,  Kong, Q.,  Mao, W., & Tang, S. | **2018** | Word embedding technology using Global vector of word representation | Top news will be ranked according to the user query | The model is trained on the cooccurrence matrix of words, which takes a lot of memory for storage |
| **4** | **Exploring mobile news reading**  **interactions for news app personalisation** | Marios  Constantinide s, John  Dowell,  David  Johson,  Sylvain  Malacria | **2015** | 1. Identification of news reader types 2. Interaction logging and classification study 3. Deployment and data collection 4. Predicting News reader types  5. Adaptive UI | The adaptive user interface changes according to the type of task you want to perform. This will increase the stability of  the system. | The overall code of the website and the size of the app increases if the user interface is adaptive. There is a lot of code to be written for making the user interface adaptive |
| **5** | **Detection and Tracking in**  **News Articles** | Sagar Patel,  Sanket  Suthar,  Sandip Patel, Neha Patel | **2015** | 1. Pre-processing 2. Tokenization 3.  Stemming/Lemmizat -ion 4. Vector Space Model 5. Topic tracking | Allows the computing for a continuous degree of similarities between queries and document | 1. Suffers from synonym and polysemy 2. It theoretically assumes that terms are statistically independent. |
| **6** | **Following the**  **Fed with a News Tracker** | Michael  William  McCracken | **2012** | The paper is not a technical paper but is essentially a statistical paper on how should one conclude whether the data have come in stronger, weaker or as expected. This is based on the Citi Group U.S Economic Surprise Index. | Tracks whether a core set of economic data series has been coming in under  expectations, at  expectations, or over expectations | , the surprise indexes tend to have the right sign and be significant: a positive change in the U.S. surprise  index (i.e. the U.S. economy doing better than expected) appreciates the U.S. dollar versus the foreign currency, whereas a positive change in the foreign surprise index depreciated the U.S. dollar. |

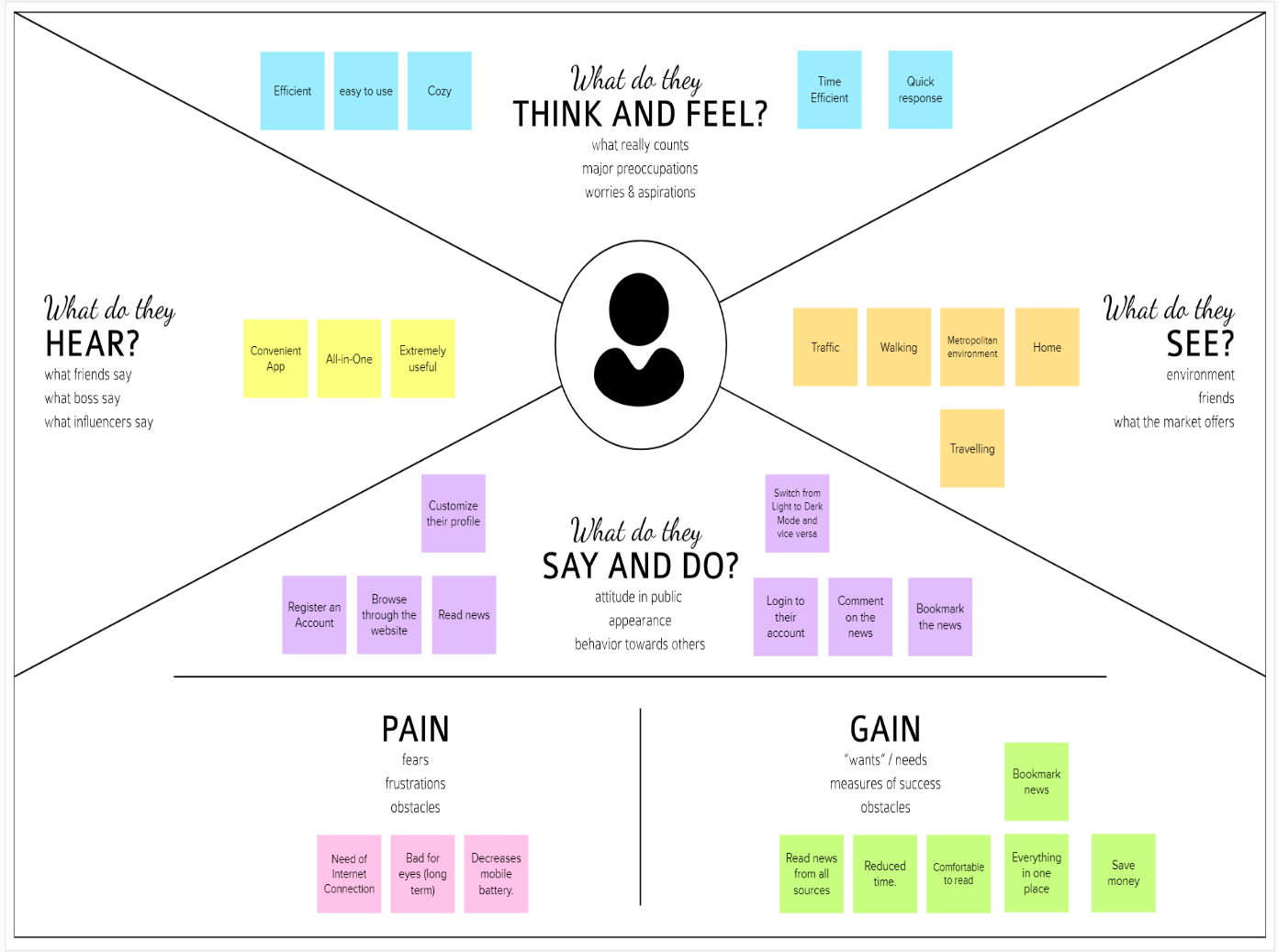
## 2.3 Problem Statement Definition

Newspaper contains limited, nonuser/reader specific, Location specific news. There are multiple news-sharing apps available which can be used by a single user and are often spammed with notifications. There is also a lot of unwanted news which gets shared. So it may take a lot of time for the user to find the news he/she likes. 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.

Vijay is a busy business man who needs to read news on the go without any hassles while travelling because he considers carrying around a physical newspaper a nuisance to him and the people around him. Vijay needs to read needs to read news in such a way that he doesn't have to worry about ever buying physical newspapers or carrying with him everywhere. Something which should fit in the palm of his hands, which he could carry everywhere, access from everywhere, something digital such as an Application hosted on the internet which could be accessed from any device that is connected to the Internet. Such as smartphones and computers.

# 3. IDEATION & PROPOSED SOLUTION

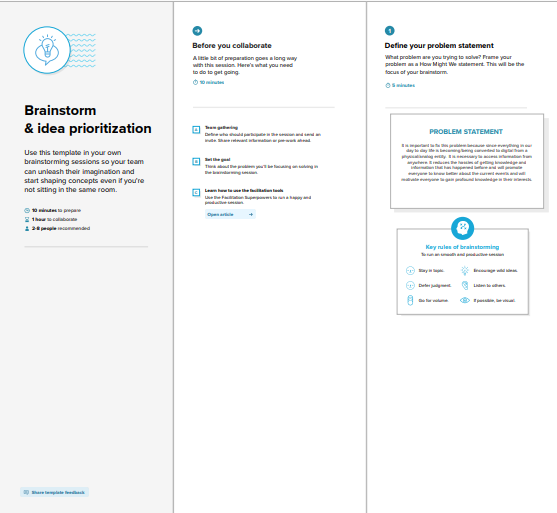
## 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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



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

Graphical user interface, application

Description automatically generated

## Step-3: Idea Prioritization

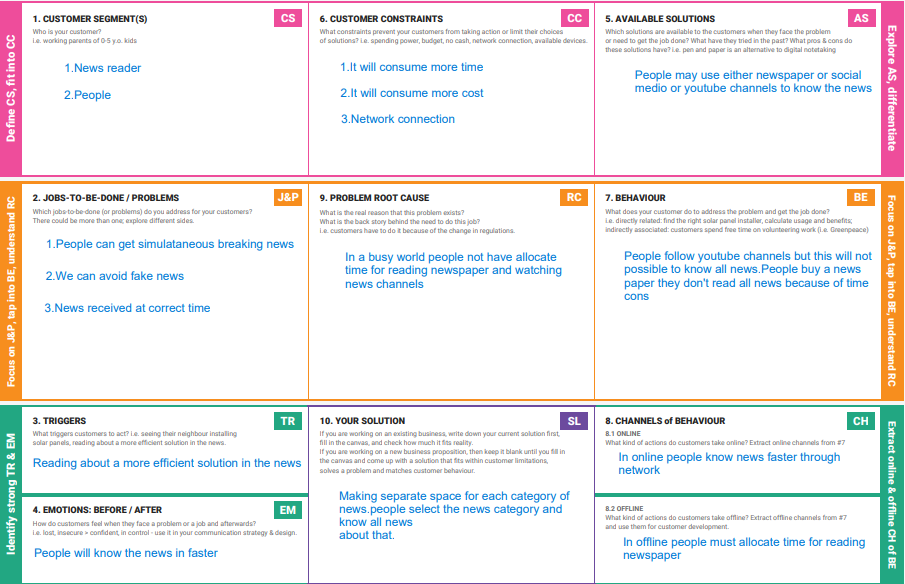
Chart

Description automatically generated

**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 information in an efficient way.    Our platform eliminates the spread of false news and exposes the injustice and wrongdoings done by false groups.    Eliminating the fake news provides better understanding of the real-events happening in the world and the spread of knowledge. |
| **5.** | Business Model (Revenue Model) | Our application covers a range of topics including politics, business, criminal justice, environment, technology etc.    Our business model will be monetized and generate income by showing advertisements and Operating on monthly and yearly subscription model. |
| **6.** | Scalability of the Solution | Scalability is one of the benchmarks of the cloud services and its adoption with businesses.    Cloud scalability will help to increase the user- base by increasing the resource allocation and meeting the changing demands without sacrificing the efficiency or quality of our customer service and internal operations.    Providing fast and reliable news while maintaining positive relationships with your customers. |

## 3.4 Problem Solution fit



**4. 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 Registration | Registration through online application  Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | User login | Login through browser directly by entering username and password  Login through  Login through email |
| FR-4 | User interaction | Done through user interface between client and server  View the related news by subscripted or requested page |

**Non-functional Requirements:**

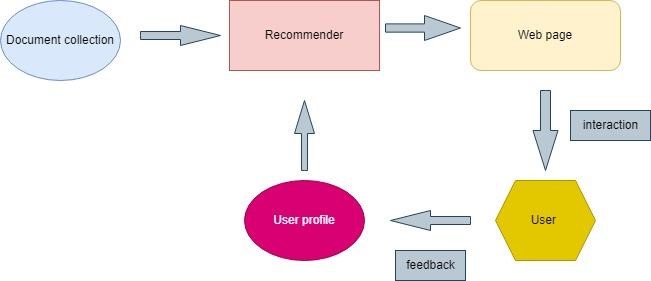
Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Non-Functional**  **Requirement** | **Description** |
| NFR-1 | **Usability** | End users can receive push updates for new content on a site by subscribing to the site’s news feed |
| NFR-2 | **Security** | How well are the system and its data protected against attacks |
| NFR-3 | **Reliability** | How often does the system experience critical failures? How much time does it take to fix the issue when it arises ?And how is user availability time compared to downtime? |
| NFR-4 | **Performance** | Performance is the core non-functional requirements no system can do without. It defines how fast a software system or a particular piece of it responds to certain users actions under a certain workload. In most cases, this metric explains how long a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment.  But it’s not always like that. Performance requirements may describe background |
|  |  | processes invisible to users, e.g. backup.  But let’s focus on user-centric performance. |
| NFR-  5 | **Availability** | Availabilitydescribes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period. For instance, the system may be available 98 percent of the time during a month. Availability is perhaps the most [business-critical requirement,](https://www.altexsoft.com/blog/business-requirements-document/) but to define it, you also must have estimations for reliability and maintainability. |
| NFR-  6 | **Scalability** | Scalabilityassesses the highest workloads under which the system will still meet the performance requirements. There are two ways to enable your system scale as the workloads get higher: horizontal and vertical scaling. |

## 5. PROJECT DESIGN

### 5.1 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.



Diagram

Description automatically generated

### 5.2 Solution & Technical Architecture

Graphical user interface, diagram

Description automatically generated

### **5.3 User Stories**

### 

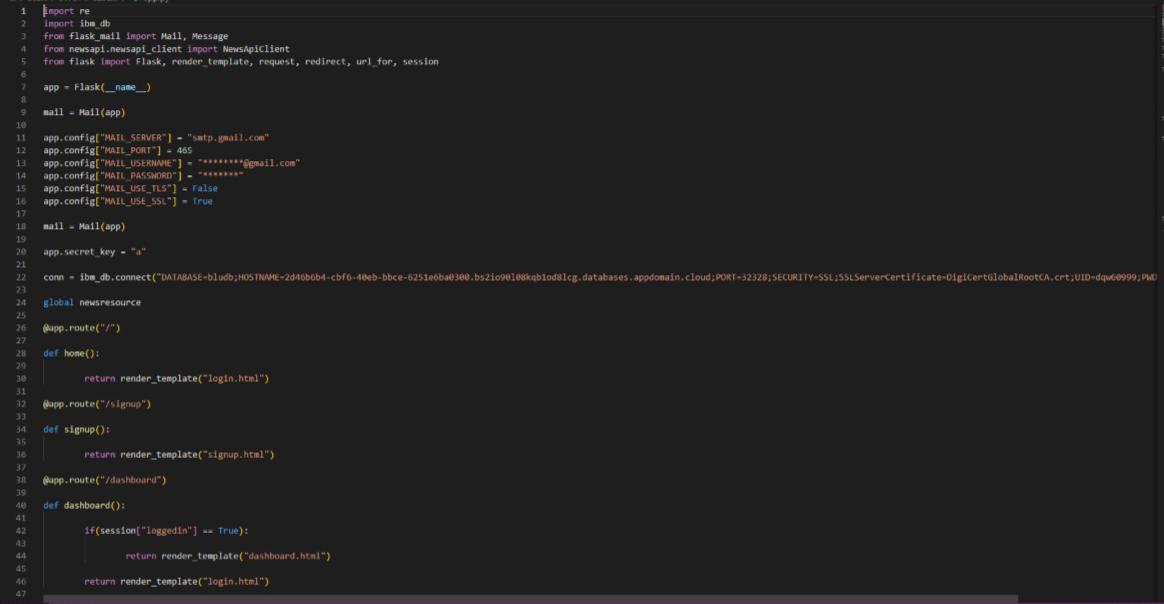
## 6. PROJECT PLANNING & SCHEDULING

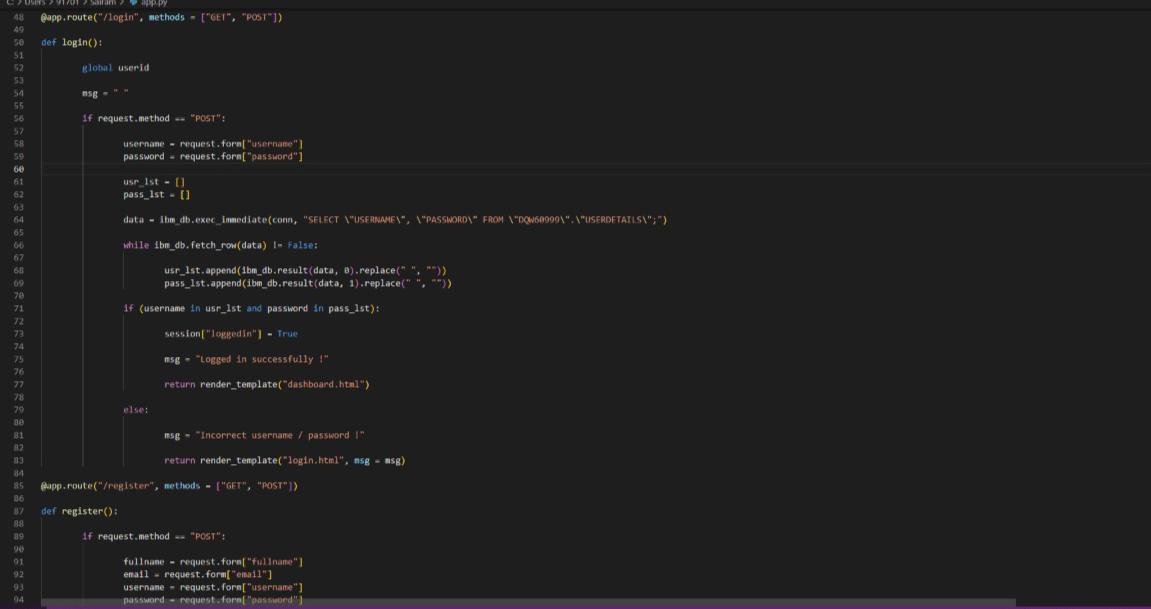
### **6.1 Sprint Planning & Estimation**

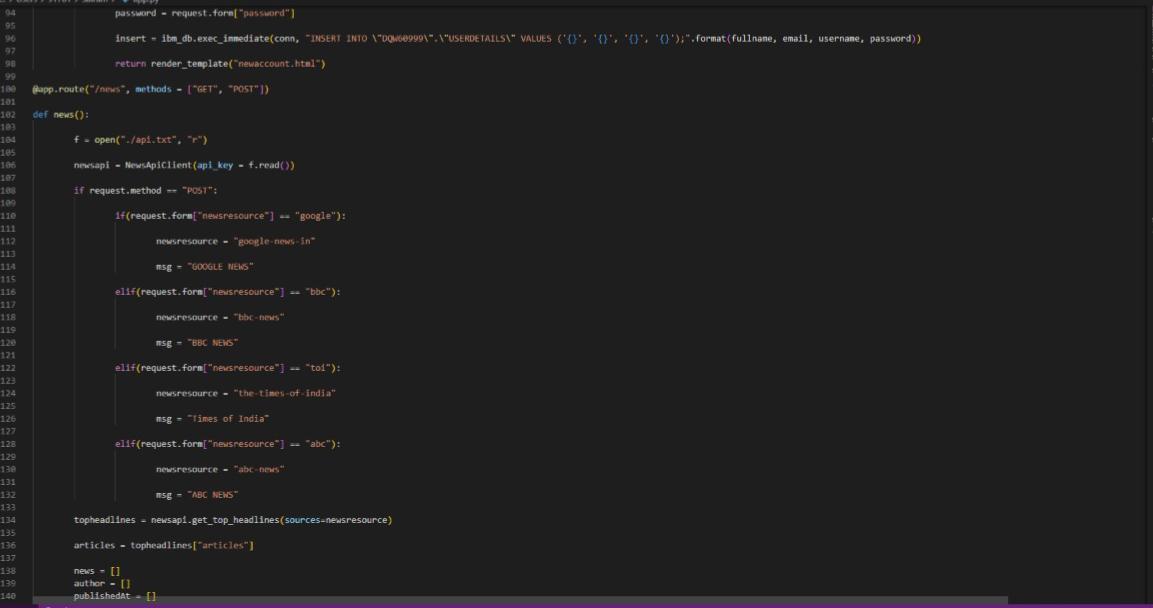
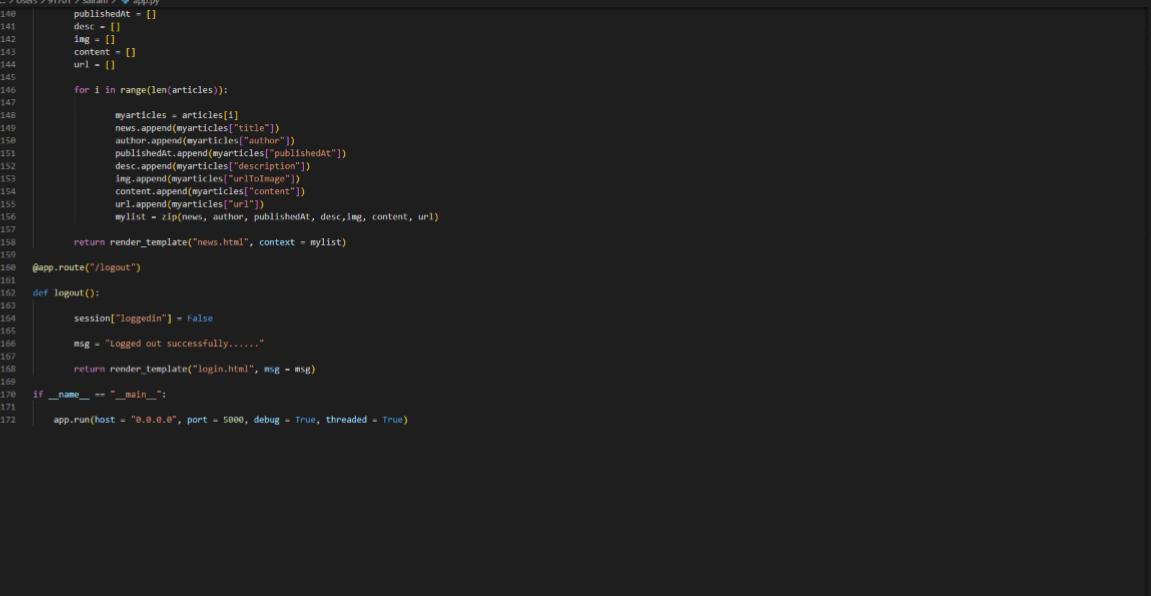
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (EPIC)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| **Sprint - 1** | Registration | USN - 1 | As a Number, I can Register for the application by entering my password.as a User, I will receive confirmation email once I have Registered for the application | 10 | HIGH | MONISH V, MANIGANDAN R |
| **Sprint - 1** | Login | USN - 2 | As a user, I will receive confirmation email once I have registered for the application | 5 | HIGH | SAIRAM G, ROHITH B |
| **Sprint - 1** | Email Verification | USN - 3 | As A User I Can Verify My Email Using the Link Sent to My Mail id | 5 | HIGH | MONISH V, SAIRAM G |
| **Sprint - 2** | API Fetch | USN - 4 | Fetch News from Rapid API at regular interval | 10 | HIGH | MANIGANDAN R, ROHITH B |
| **Sprint - 2** | REST Endpoints for Backend | USN - 5 | Creating endpoints at the backend in order to interact with frontend | 10 | MEDIUM | MANIGANDAN R, SAIRAM G |
| **Sprint - 3** | Designing Frontend | USN - 6 | Create a minimalisting design in figma to create frontend | 2 | MEDIUM | MONISH V, ROHITH B |
| **Sprint - 3** | Creating Frontend | USN - 7 | Create the frontend webpage using the design | 10 | LOW | SAIRAM G, ROHITH B |
| **Sprint - 3** | Connect Frontend and backend | USN - 8 | Connect the frontend and backend and complete the application | 8 | HIGH | MONISH V, MANIGANDAN R |
| **Sprint - 4** | Testing | USN - 9 | Testing the application before final release | 10 | HIGH | SAIRAM G, MONISH V |
| **Sprint - 4** | Deployment | USN - 10 | Deployment of the application | 10 | HIGH | MANIGANDAN R, ROHITH B |

### **6.2 Sprint Delivery Schedule**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Milestone List | **OCTOBER** | | | | | | | | | | | **NOVEMBER** | | | | | | | | | | | | | | | |
| **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **01** | | **02** | **03** | **04** | **05** | **06** | **07** | **08** | **09** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **Sprint 1- Registration and sign in** |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design sign up & sign in page | 1 DAY |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Email Auth |  | 2 DAY | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DB2 Database Design |  |  | 1 DAY |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Email & Password sign in |  |  |  | 1 DAY |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Email Configuration on user account creation |  |  |  |  | 1 DAY |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Sprint 2 – API Fetching and Backend endpoints** |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fetch data from rapid API |  |  |  |  |  | 2 DAY | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flask-REST API coding |  |  |  |  |  |  |  | 2 DAY | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Create timed function for fetch from API using threading |  |  |  |  |  |  |  |  |  | 1 DAY |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test backend |  |  |  |  |  |  |  |  |  |  | 2 DAY | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Sprint 3 – UI and UX design and connecting frontend and backend** |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design main welcome page |  |  |  |  |  |  |  |  |  |  |  | |  | 3 DAY | | |  |  |  |  |  |  |  |  |  |  |  |
| News card design |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | 2 DAY | |  |  |  |  |  |  |  |  |  |
| Explore designs and saved design |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  | 1 DAY |  |  |  |  |  |  |  |  |
| Bookmarks design |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | 1 DAY |  |  |  |  |  |  |  |
| Connecting frontend and backend |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  | 1 DAY |  |  |  |  |  |  |
| UI responsiveness |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  | 2 DAY | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Sprint 4- Deployment, testing and integrator** |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Deploying the App on cloud |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | 3 DAY | | |  |  |
| Implementing loggers |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | 2 DAY | |  |
| Final testing |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 DAY | |

**7. CODING**





## OUTPUT

## 

## 

## 

## 

## 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

* [*Source Code (Github)*](https://github.com/IBM-EPBL/IBM-Project-35054-1660281098)