

## **Final Report**

### **NEWS TRACKER APPLICATION**

#### **Team Members:**

**R SIDHARTH**

**P GOVARDHAN**

**D SRABESWARAN**

**S VASANTHARAJA**

# **Project Report Format**

## **1. INTRODUCTION**

1.1. Project Overview

1.2. Purpose

## **2. LITERATURE SURVEY**

2.1. Existing problem

2.2. References

2.3. Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

3.1. Empathy Map Canvas

3.2. Ideation & Brainstorming

3.3. Proposed Solution

3.4. Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

4.1. Functional requirement

4.2. Non-Functional requirements

## **5. PROJECT DESIGN**

5.1. Data Flow Diagrams

5.2. Solution & Technical Architecture

5.3. User Stories

## **6. PROJECT PLANNING & SCHEDULING**

6.1. Sprint Planning & Estimation

6.2. Sprint Delivery Schedule

6.3. Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

7.1. Feature 1

7.2. Feature 2

7.3. Database Schema (if Applicable)

## **8. TESTING**

8.1. Test Cases

8.2. User Acceptance Testing

## **9. RESULTS**

9.1. Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# **1. INTRODUCTION**

## **1.1 Project Overview:**

The project is to build a NEWS tracker application where different users can view the news what they prefer instead of showing all news. The application has been build using flask, html, CSS and JavaScript. This application has been linked with IBM cloud database and also has IBM Watson chatbot integration. This provides the user a good user interaction and also makes the application easier to use.

## **1.2 Purpose:**

Tracking news gives important information to end users to keep track of latest facts and information. Specific users need specific news for their needs like economic news for business people, weather news for fisherman, etc. So, it becomes necessary to track specific news for the intended user. As news is increasingly accessed on smartphones and tablets, the need for personalizing news app interactions is apparent. So, the project aim is to build a news tracker application where users can able to track specific news for their needs along with various additional functionalities.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

- Reviewers reported they did not find the adaptive menu beneficial and would prefer a snapshot of articles within multiple categories as opposed to being restricted to one. None reported wanting article summaries despite reading long articles and being probed
- Algorithm-based recommendation of this cutting-edge technology will face numerous tests from journalists, social ethics, laws and regulations.
- The importance of trust has been highlighted future research studies can focus on the antecedents of trust Users require lower level of trust.

### **2.2 References**

1. Tracking sentiment towards news entities from Arabic news on social media - Ali Al-Laith, Muhammad Shahbaz - Future generation computer system.
2. Exploring mobile news reading interactions for news app Personalization - Marios Constantinides, John Dowell, David Johnson, Sylvain Malacria - 17th International Conference on Human computer Interaction with Mobile Devices and Services.
3. Research on Development Strategy of News App under the Background of Artificial Intelligence -Wei Guo and Bo Zhang - International Conference on AI and Big Data Application (AIBDA 2019).
4. Topic Detection and Tracking in News Articles - Sagar Patel, Sanket Suthar, Sandip Patel, Nehal Patel and Arpita Patel Chandubhai S - Information and Communication Technology for Intelligent Systems.

### **2.3 Problem Statement Definition**

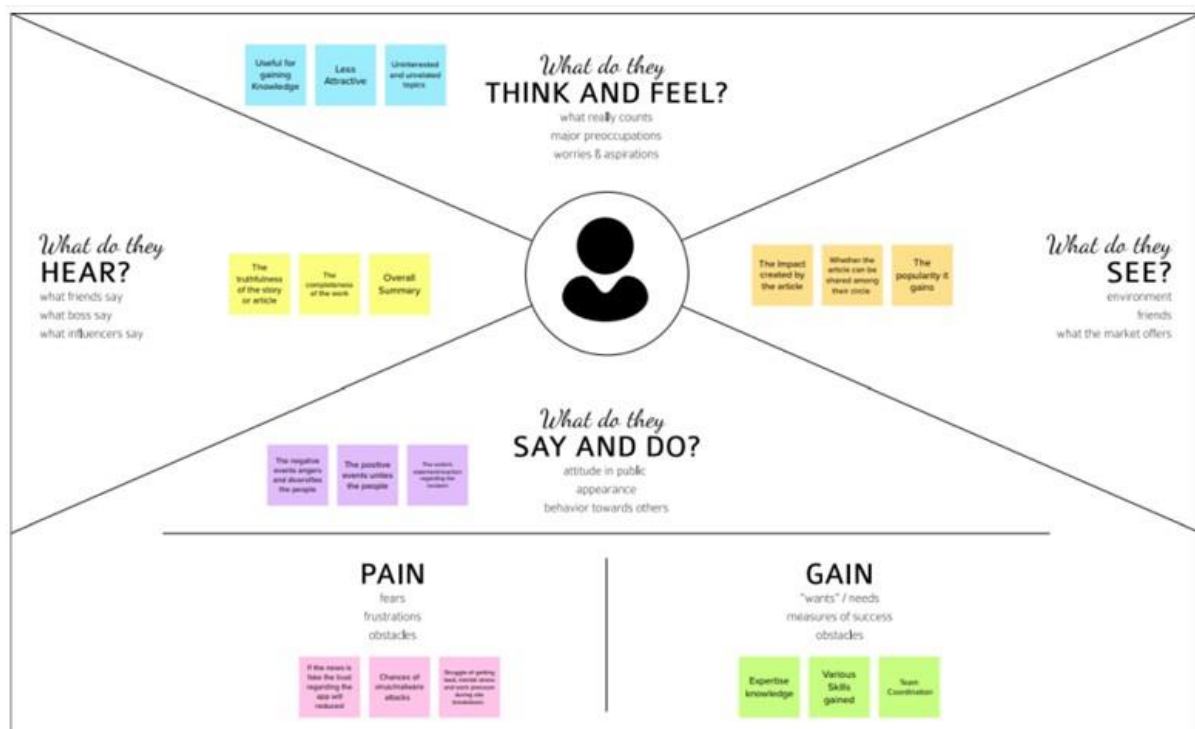
News reading is the basic source of gaining knowledge about the external growing world. People read newspapers to gain knowledge about what is happening around the world. Without newspaper people cannot be able to educate themselves about the outside world. In Modern Era this kind of culture changed, and Television and Radio are used as a better source of communication and Knowledge gathering tool. Today there are several easy to educate ourselves, one of the easier ways is watching news in our mobile phone. The main aim is to develop a web application to track news and ongoing news feeds and displaying them.

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.


##### Empathy Map:



#### 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 number of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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



### Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

⌚ 10 minutes to prepare  
🕒 1 hour to collaborate  
👥 2-8 people recommended

**Before you collaborate**  
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

⌚ 10 minutes

**A Team gathering**  
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

**B Set the goal**  
Think about the problem you'll be focusing on solving in the brainstorming session.

**C Learn how to use the facilitation tools**  
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

**1 Define your problem statement**  
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⌚ 5 minutes

**PROBLEM**

How might we [your problem statement]?

**Key rules of brainstorming**  
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

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

**1 Brainstorm**  
Write down any ideas that come to mind that address your problem statement.

⌚ 10 minutes

**2 Group ideas**  
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence or two label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

⌚ 20 minutes

**Venkateshwar S**

|  |   |   |
|--|---|---|
| Checklist                                    | Read Offline                                    | History   |
| User Interface is interactive and catchy     | Translate the NEWS into Hindi language          | Ads to add only inside the users                        |
| Recommendation and group news as a grid view | Make short-cut to report live news from the app | Use social media to share news and connect with friends |

**Vignesh B**

|  |  |   |
|--|--|---|
| Comments and share news to others              | News in 30 words   | Reduce Spam Notifications   |
| Recommendation NEWS by user location           | Personalization news by location, news in Hindi, News Personal | Generate income through subscription                                      |
| Use short-cut to report live news from the app | People prefer personalized push notifications                  | People want to see news frequently in their feeds and not cluttered feeds |

**Vishal Kanna AJK**

|   |   |   |
|---|---|---|
| Feedback                                    | Feeding Of Generated NEWS                         | Storage   |
| Collection, Feedback, Recommendation Filter | Integrate with social media and connect the users | Comparison Analysis                                     |
| News in 30 words                            | Make short-cut to report live news from the app   | Use social media to share news and connect with friends |

**Abhishek S**

|                                    |   |   |
|------------------------------------|---|---|
| Mobile App Accessibility           | Generation Of News NEWS                       | Pushes short stories with NEWS articles                                   |
| Personalized NEWS by user location | Push NEWS                                     | App also should be small  |
| News in 30 words                   | People prefer personalized push notifications | People want to see news frequently in their feeds and not cluttered feeds |

**CUSTOMIZATION:**

|  |                           |   |                                   |  |   |                          |
|--|---------------------------|---|-----------------------------------|--|---|--------------------------|
| Translate the NEWS into Hindi language | Feedback                  | People prefer personalized push notifications | Generate and share news to others | User Interface is interactive and catchy | Collection, Feedback, Recommendation Filter | App also should be small |
| Feeding Of Generated NEWS              | Reduce Spam Notifications |   |                                   |  |   |                          |

**INNOVATION IDEAS:**

|                                   |   |   |                  |              |          |                          |
|-----------------------------------|---|---|------------------|--------------|----------|--------------------------|
| Personalized NEWS can be notified | Pushes short stories with NEWS articles | Integrate with social media and connect the users | News in 30 words | Read Offline | Checkout | Mobile App Accessibility |
|-----------------------------------|---|---|------------------|--------------|----------|--------------------------|

**NECESSARY IDEAS:**

|   |   |   |                  |              |                          |                          |
|---|---|---|------------------|--------------|--------------------------|--------------------------|
| App also should be small                | Comparison Analysis                               | Integrate with social media and connect the users | News in 30 words | Read Offline | Checkout                 | Mobile App Accessibility |
| Pushes short stories with NEWS articles | Integrate with social media and connect the users | News in 30 words                                  | Read Offline     | Checkout     | Mobile App Accessibility |                          |

## Step-3: Idea Prioritization

### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

### After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

**Quick add-ons**

- Share the mural**  
Share a view link to the mural with collaborators to keep them in the loop about the outcomes of the session.
- Export the mural**  
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save on your drive.

**Keep moving forward**

- Strategy blueprint**  
Define the components of a new idea or strategy.  
[Open the template](#)
- Customer experience journey map**  
Understand customer needs, motivations, and obstacles for an experience.  
[Open the template](#)
- Strengths, weaknesses, opportunities & threats**  
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.  
[Open the template](#)

[Share template feedback](#)

## 3.3 Proposed Solution

| S.No. | Parameter                                | Description  |
|-------|--|--|
| 1.    | Problem Statement (Problem to be solved) | News reading is the basic source of gaining knowledge about the external world. People read newspapers to gain knowledge about what is happening around the world. Today there are several ways to educate ourselves, one of the easier ways is reading news in our mobile phone. So developing an application for that becomes necessary. |
| 2.    | Idea / Solution description              | The idea is to develop a application that tracks specific news for individual users using flask.   |
| 3.    | Novelty / Uniqueness                     | Our cloud-based NEWS tracker application has the following uniqueness: <ul style="list-style-type: none"> <li>1. It is fast and cost-effective.</li> <li>2. It eliminates the risk of human errors during expense calculation when manually.</li> </ul>  |

|    |                                       |   |
|----|---------------------------------------|---|
|    |                                       | <ul style="list-style-type: none"> <li>3. It is more secure.</li> <li>4. It offers better analytics and transparency.</li> </ul>  |
| 4. | Social Impact / Customer Satisfaction | Making an application to track NEWS helps the user to find specific NEWS what they want rather than searching it from a vast source   |
| 5. | Business Model (Revenue Model)        | We can provide the application on a subscription and advertising basis and the cost depends on the usage.   |
| 6. | Scalability of the Solution           | <ul style="list-style-type: none"> <li>a. Instead of switching to a bigger instance size, our cloud-based application uses horizontal scalability to supply more instances, and flask micro services are utilized to improve specific functionalities.</li> <li>b. This application can handle large number of users and data with high performance and security</li> </ul> |



## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task)  |
|--------|-------------------------------|---|
| FR-1   | User Registration             | <ol style="list-style-type: none"><li>1. Registration through Form</li><li>2. Registration through Gmail</li><li>3. Registration through LinkedIn</li></ol>   |
| FR-2   | User Confirmation             | <ol style="list-style-type: none"><li>1. Confirmation via Email</li><li>2. Confirmation via OTP</li></ol>   |
| FR-3   | Searching                     | <ol style="list-style-type: none"><li>1. Search the news based on the user interest</li><li>2. User can save the news needed and study later</li><li>3. Showing the trending news on search</li></ol>   |
| FR-4   | Real time News                | <ol style="list-style-type: none"><li>1. Auto update news because it fetches news from API</li><li>2. Use of ML algorithm for up-to-date information</li></ol>  |
| FR-5   | Location based                | <ol style="list-style-type: none"><li>1. User can view the news near to their location</li><li>2. User can track the Location of the news</li></ol>   |
| FR-6   | User friendly                 | <ol style="list-style-type: none"><li>1. Push notification are meant to attract user attention by using alerts, even if it is just for a second.</li><li>2. Links provided to get more information on the related news.</li><li>3. User can get the required news by searching the context.</li></ol> |

## 4.2 Non-Functional requirements

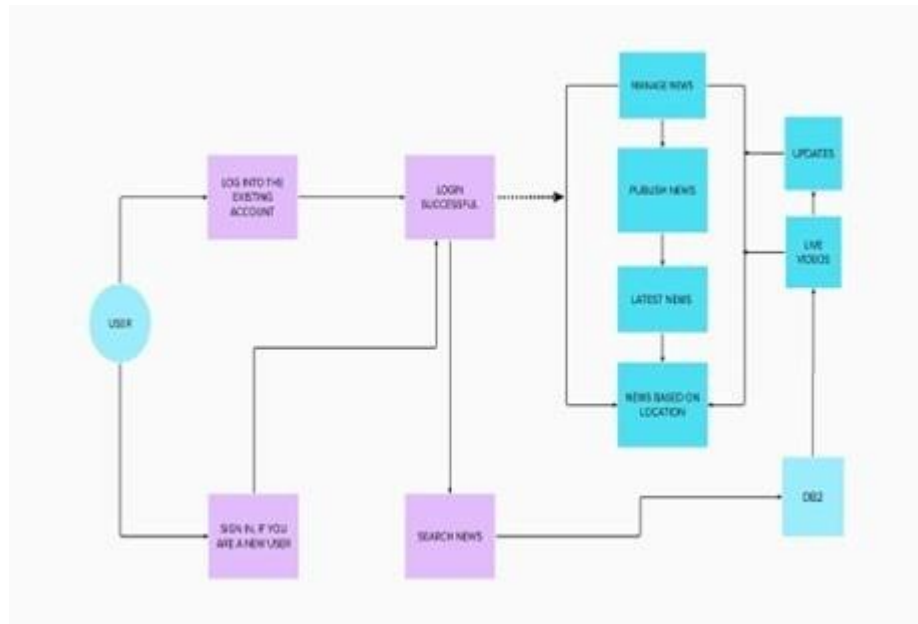
Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description   |
|--------|----------------------------|---|
| NFR-1  | <b>Usability</b>           | <ol style="list-style-type: none"><li>1. The news is updated almost every minute, the user will always be able to view exact information that interest him/her in the moment.</li><li>2. Search and Filtering function will provide the users the opportunity to search for information of their interest</li></ol> |
| NFR-2  | <b>Security</b>            | <ol style="list-style-type: none"><li>1. Authentication and password management.</li><li>2. Personal profile is not to shared to thirdparty.</li></ol>  |
| NFR-3  | <b>Reliability</b>         | <ol style="list-style-type: none"><li>1. Instant news at instant time</li><li>2. Track the location of the news</li><li>3. Avoid the Fake news</li></ol>  |
| NFR-4  | <b>Performance</b>         | <ol style="list-style-type: none"><li>1. Show users the most relevant story first</li><li>2. Recommended news based on user likings</li><li>3. Keep users in the app for longer with related post</li></ol>   |
| NFR-5  | <b>Availability</b>        | <ol style="list-style-type: none"><li>1. Provides 24/7 news service</li><li>2. History of the previous news that happened before related to the present news</li></ol>  |
| NFR-6  | <b>Scalability</b>         | <ol style="list-style-type: none"><li>1. Get more user's by encouraging social sharing and recommending the app to new users.</li><li>2. Getting user feedback and make changes to keep the customer stay with the application.</li></ol>   |

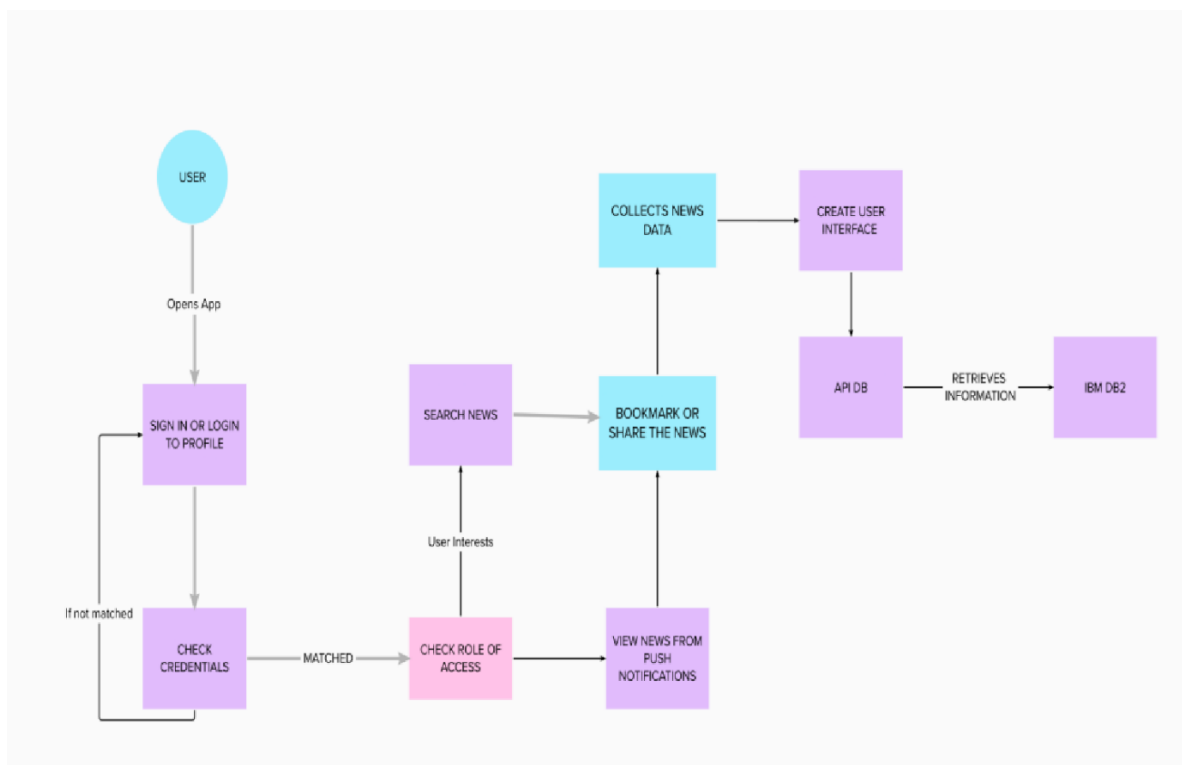
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

#### Example (Simplified)



#### Example: DFD Level 0 (Industry Standard)



## 5.2 Solution & Technical Architecture

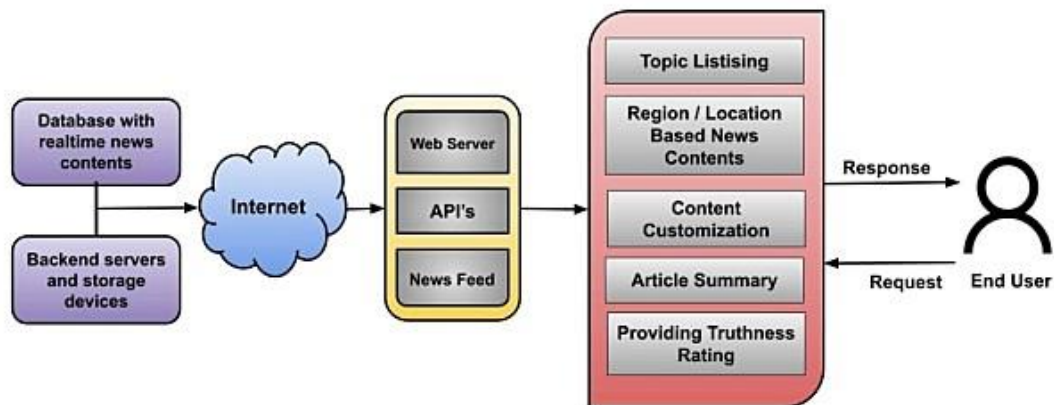
### Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

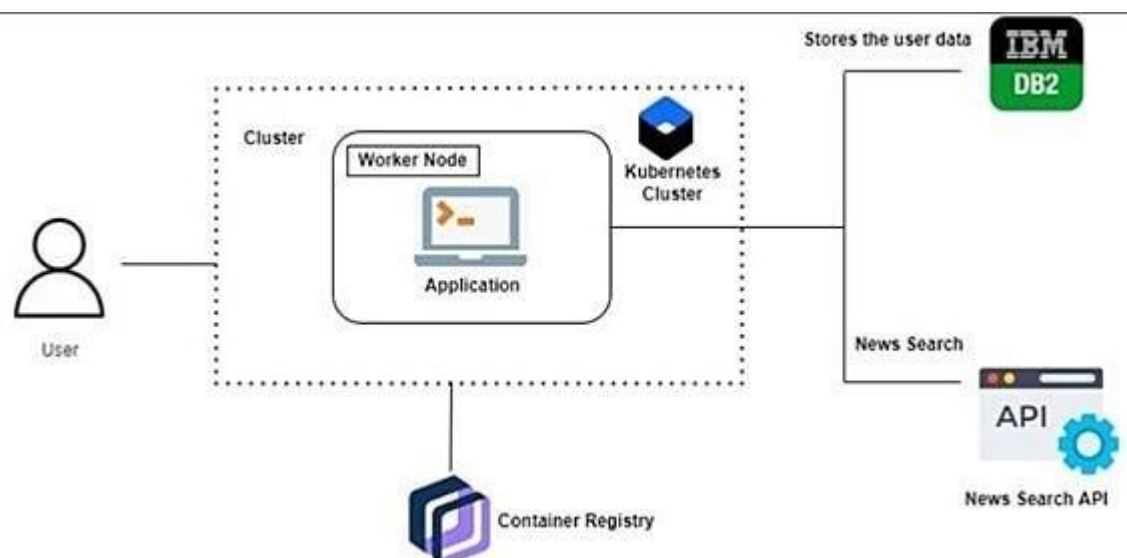
Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

### Example - Solution Architecture Diagram:



### Technology Architecture:



**Guidelines:**

1. The HTML and CSS are used for the user interface for the user to use the application.
2. The user search for the news based on interest rather than scrolling down the all the news.
3. The news feed is list of newly published content on website. End users can receive push updates for new content on a site by subscribing to site's news feed.
4. IBM DB2 on cloud is used to access the information of the news based on the user interest

**Table-1: Components & Technologies:**

| S.No | Component                       | Description  | Technology                           |
|------|---------------------------------|--|--------------------------------------|
| 1.   | User Interface                  | The User interacts with the application via an web application to read news.   | HTML, CSS, JavaScript ,React Js etc. |
| 2.   | Application Logic-1             | Verifying the client ID.   | Java / Python                        |
| 3.   | Application Logic-2             | Analyse the news feed for each customer.   | Flask                                |
| 4.   | Application Logic-3             | Integrating the customer feed with IBM Watson to improve user experience.  | IBM Watson Assistant                 |
| 5.   | Database                        | The client information is stored in a database.  | MySQL, NoSQL, etc.                   |
| 6.   | Cloud Database                  | The news feed data is stored in cloud.   | IBM DB2.                             |
| 7.   | File Storage                    | The files which the user uploads is stored here.   | IBM Object Storage.                  |
| 8.   | External API-1                  | Weather API is used to gather weather information.   | IBM Weather API, etc.                |
| 9.   | External API-2                  | To verify the identity of user we use Aadhar API   | Aadhar API, etc.                     |
| 10.  | Machine Learning Model          | ML model is used to filter the news required by the user.  | Object Recognition Model, etc.       |
| 11.  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud<br>Local Server Configuration:<br>Docker<br>Cloud Server Configuration : Kubernetes | Local, Kubernetes, etc.              |

**Table-2: Application Characteristics:**

| S.No | Characteristics          | Description   | Technology   |
|------|--------------------------|---|--|
| 1.   | Open-Source Frameworks   | A software for which the original source code is made freely available and may be redistributed and modified according to the requirement of the user.  | Flask, Docker, etc.  |
| 2.   | Security Implementations | Cloud Security Posture Management(CSPM), Detect cloud security and compliance configuration risk, anomalous activity,   | Built-in encryption, BYOK  |
| 3.   | Scalable Architecture    | Python is one of the pioneers of programming languages that developers can use to do all the scaling work. To improve scalability, you can enable or disable services run by the dispatcher on individual servers to balance the load for a given computer by request type.   | Technology used in the architecture is that with the Python and the IBM cloud. |
| 4.   | Availability             | Availability is the ability of a system to withstand or recover from exceptional situations, such as a computer failure. IBM Cloud is on-demand access, via the internet, to computing resources applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more hosted at a remote data centre managed by a cloud services provider (or CSP). | Technology used are the IBM cloud and the database.                            |
| 5.   | Performance              | DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyse and retrieve the data efficiently. DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.  | Technology used are the python, cloud and IBM db2.                             |

### 5.3 User Stories

| User Type              | Functional Requirement (Epic) | User Story Number | User Story / Task   | Acceptance criteria                                       | Priority | Release  |
|------------------------|-------------------------------|-------------------|---|---|----------|----------|
| Customer (Mobile user) | Registration                  | USN-1             | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard                       | High     | Sprint-1 |
|                        |                               | USN-2             | As a user, I will receive confirmation email once I have registered for the application                   | I can receive confirmation email & click confirm          | High     | Sprint-1 |
|                        |                               | USN-3             | As a user, I can register for the application through Facebook  | I can register & access the dashboard with Facebook Login | Low      | Sprint-2 |
|                        |                               | USN-4             | As a user, I can register for the application through Gmail   | I can receive password from mail.                         | Medium   | Sprint-1 |
|                        | Login                         | USN-5             | As a user, I can log into the application by entering email & password                                    | I can receive confirmation mail.                          | High     | Sprint-1 |
|                        | Dashboard                     | USN-6             | The news portal fetches for the most recent news and shows it as "Breaking News".                         | I can open and view the "Breaking News"                   | Low      | Sprint-2 |

|                        |            |        |  |   |        |          |
|------------------------|------------|--------|--|---|--------|----------|
| Customer<br>(Web user) | Search Bar | USN-7  | User search data based on their personal interests   | I can view the related content for the search data            | High   | Sprint-1 |
|                        |            | USN-8  | The news is showed with a sample snapshot and mini description and user can open and read the article. | I can view the news details and can open to see whole article | High   | Sprint-1 |
| Administrator          | Server     | USN-9  | Provides the exact news from the database.   | I can see the latest news getting refreshed.                  | Medium | Sprint-1 |
|                        |            | USN-10 | Provides live news with video and audio content.   | I can see news much clearly with this feature                 | High   | Sprint-1 |



## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Project Tracker, Velocity & Burndown Chart:

| Sprint   | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|----------|--------------------|----------|-------------------|---------------------------|---|------------------------------|
| Sprint-1 | 20                 | 6 Days   | 22 Oct 2022       | 29 Oct 2022               | 29 Oct 2022                                     | 29 Oct 2022                  |
| Sprint-2 | 20                 | 6 Days   | 29 Oct 2022       | 05 Nov 2022               | 05 Nov 2022                                     | 05 Nov 2022                  |
| Sprint-3 | 20                 | 6 Days   | 05 Nov 2022       | 12 Nov 2022               | 12 Nov 2022                                     | 12 Nov 2022                  |
| Sprint-4 | 20                 | 6 Days   | 12 Nov 2022       | 19 Nov 2022               | 19 Nov 2022                                     | 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)

$$\begin{aligned} \text{AV} &= \text{Sprint Duration} / \text{Velocity} \\ &= 7 / 15 = 0.46 \end{aligned}$$

#### 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.

| S:No | Task / Activity                              | Sprint-1 24 - 29 Oct 2022 | Sprint-2 31 Oct - 5 Nov 2022 | Sprint-3 7 - 12 Nov 2022 | Sprint-4 14 - 19 Nov 2022 |
|------|--|---------------------------|------------------------------|--------------------------|---------------------------|
| 1    | Creating a web page (Home-Page & Dashboard). |                           |                              |                          |                           |
| 2    | Creating Database and working on it.         |                           |                              |                          |                           |
| 3    | Establishing database connection.            |                           |                              |                          |                           |

|    |  |  |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|--|--|
| 4  | Creating Registration/Login Page.                |  |  |  |  |  |  |  |  |
| 5  | Creating User Login.                             |  |  |  |  |  |  |  |  |
| 6  | Implementing Admin & Publisher Panel.            |  |  |  |  |  |  |  |  |
| 7  | Integrating NEWS API.                            |  |  |  |  |  |  |  |  |
| 8  | Adding Watson Assistant.                         |  |  |  |  |  |  |  |  |
| 9  | Testing and debugging.                           |  |  |  |  |  |  |  |  |
| 10 | Creating image and containerize the application. |  |  |  |  |  |  |  |  |
| 11 | Deploying the application in IBM cloud.          |  |  |  |  |  |  |  |  |
|    |  |  |  |  |  |  |  |  |  |

## 6.2 Sprint Delivery Schedule

| Sprint   | Functional Requirement (Epic) | User Story Number | User Story / Task   | Story Points | Priority | Team Members   |
|----------|-------------------------------|-------------------|---|--------------|----------|----------------|
| Sprint-1 | Registration                  | USN-1             | As a user, I can register for the application by entering my email, password, and confirming my password. | 2            | High     | Sidharth R     |
| Sprint-1 |                               | USN-2             | As a user, I will receive confirmation email once I have registered for the application                   | 1            | High     | Govardhan P    |
| Sprint-2 |                               | USN-3             | As a user, I can register for the application   | 3            | Low      | Sarabeswaran D |

|          |            |       |  |   |        |                |
|----------|------------|-------|--|---|--------|----------------|
|          |            |       | through Facebook   |   |        |                |
| Sprint-1 |            | USN-4 | As a user, I can register for the application through Gmail            | 3 | Medium | Sidharth R     |
| Sprint-1 | Login      | USN-5 | As a user, I can log into the application by entering email & password | 3 | High   | Govardhan P    |
| Sprint-2 | Dashboard  | USN-6 | Shows the recent NEWS and Breaking NEWS.                               | 5 | Medium | Sidharth R     |
| Sprint-3 | Search Bar | USN-7 | User searches for News based on their own interest.                    | 3 | High   | Vasantharaja S |
| Sprint-4 | Server     | USN-8 | Provides correct NEWS available from the database.                     | 8 | Medium | Sarabeswaran D |
| Sprint-4 |            | USN-9 | Provide live news with video and audio content.                        | 5 | High   | Vasantharaja S |

## 6.3 Reports from JIRA

### Backlog:

The screenshot displays the JIRA Backlog for the 'NEWS TRACKER APPLICATION' project. The interface includes a top navigation bar with options like 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', and 'Apps'. A search bar and a 'Quickstart' button are also present. The main content area shows a list of tasks, each with a checkbox, an ID, a title, and a phase label. The tasks are organized into two main categories: 'PROJECT DESIGN PHASE -II' and 'PROJECT DEVELOPMENT PHASE'.

| Task ID | Task Title                                 | Phase                     |
|---------|--|---------------------------|
| NTA-40  | Technology Architecture                    | PROJECT DESIGN PHASE -II  |
| NTA-39  | Data flow diagrams                         | PROJECT DESIGN PHASE -II  |
| NTA-38  | Functional Requirement                     | PROJECT DESIGN PHASE -II  |
| NTA-37  | Customer Journey                           | PROJECT DESIGN PHASE -II  |
| NTA-36  | Solution Architecture                      | PROJECT DESIGN PHASE -I   |
| NTA-35  | Problem Solution Fit                       | PROJECT DESIGN PHASE -I   |
| NTA-34  | Proposed Solution                          | PROJECT DESIGN PHASE -I   |
| NTA-45  | Project Development - Delivery of Sprint-3 | PROJECT DEVELOPMENT PHASE |
| NTA-44  | Project Development - Delivery of Sprint-2 | PROJECT DEVELOPMENT PHASE |
| NTA-43  | Project Development - Delivery of Sprint-1 | PROJECT DEVELOPMENT PHASE |
| NTA-46  | Project Development - Delivery of Sprint-4 | PROJECT DEVELOPMENT PHASE |

## Board:

Projects / NEWS TRACKER APPLICATION

NTA Sprint 1

13 days remaining

Complete sprint

...

Q

Epic 1

Clear filters

GROUP BY: None

Insights

TO DO 1 OF 1 ISSUE

IN PROGRESS 2 OF 6 ISSUES

DONE 2 OF 20 ISSUES

Create an account in Sendgrid  
SETTING UP APPLICATION ENVIRO...  
NTA-24

Install IBM Cloud CLI  
SETTING UP APPLICATION ENVIRO...  
NTA-22

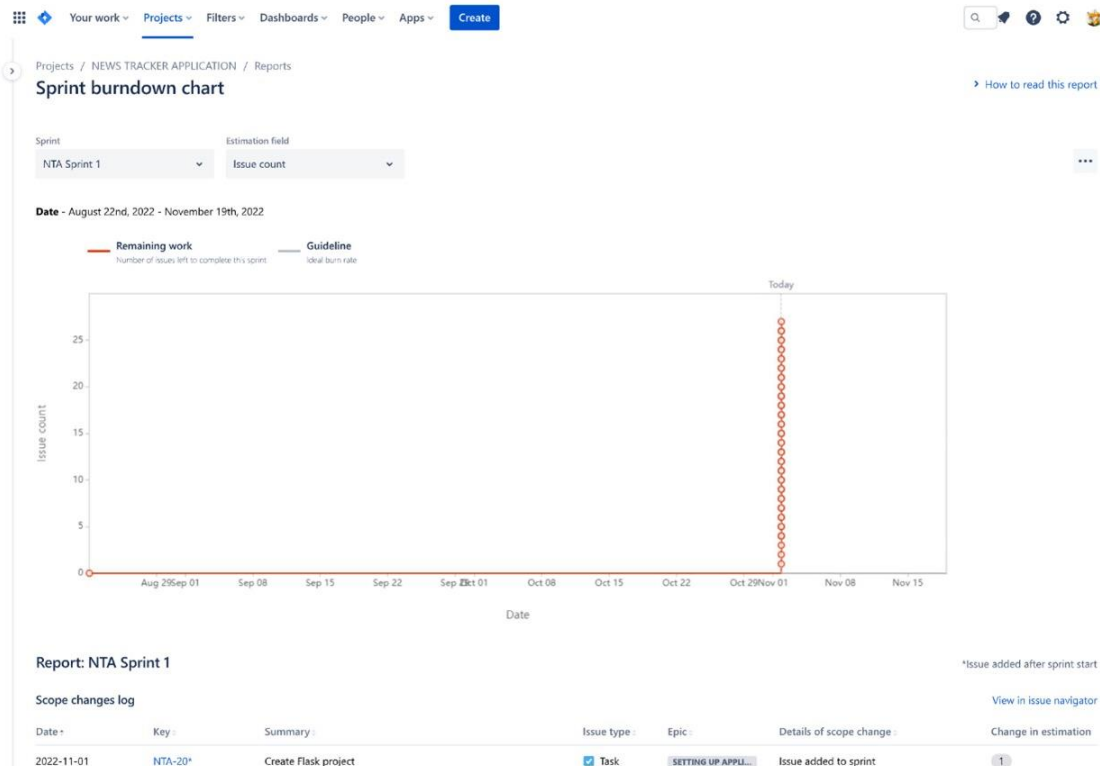
Docker CLI Installation  
SETTING UP APPLICATION ENVIRO...  
NTA-23

Create Flask project  
SETTING UP APPLICATION ENVIRO...  
NTA-20

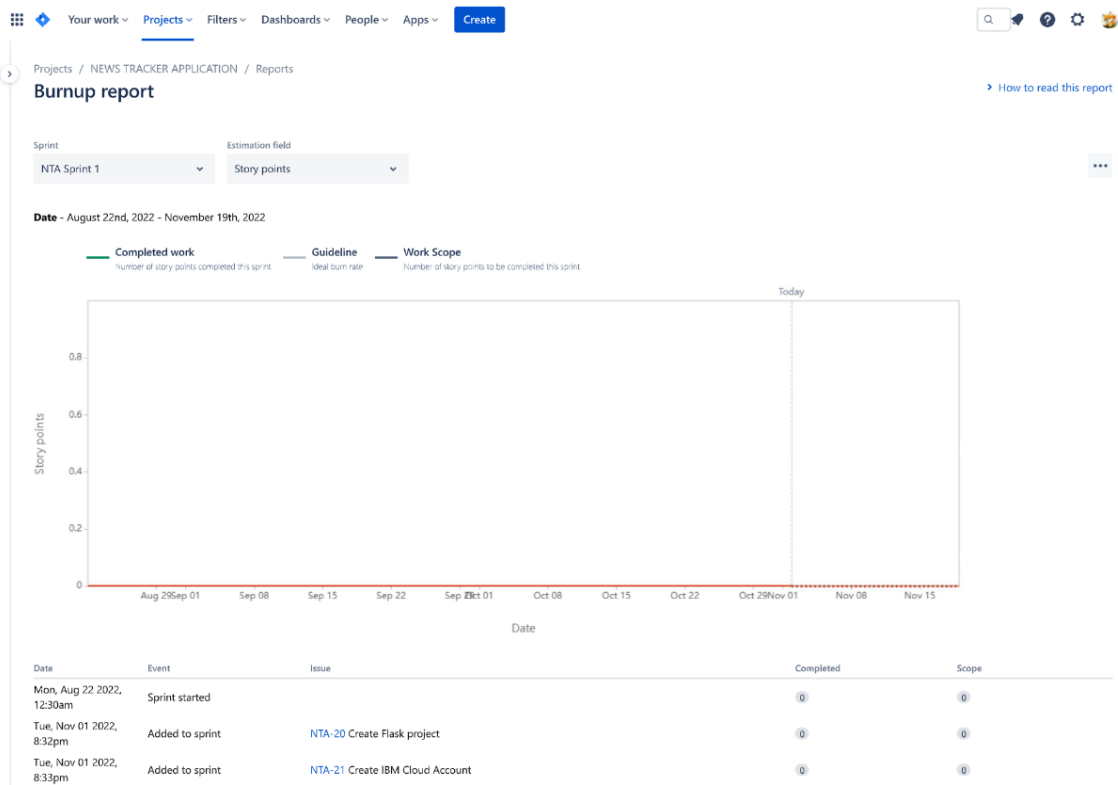
Create IBM Cloud Account  
SETTING UP APPLICATION ENVIRO...  
NTA-21

Quickstart

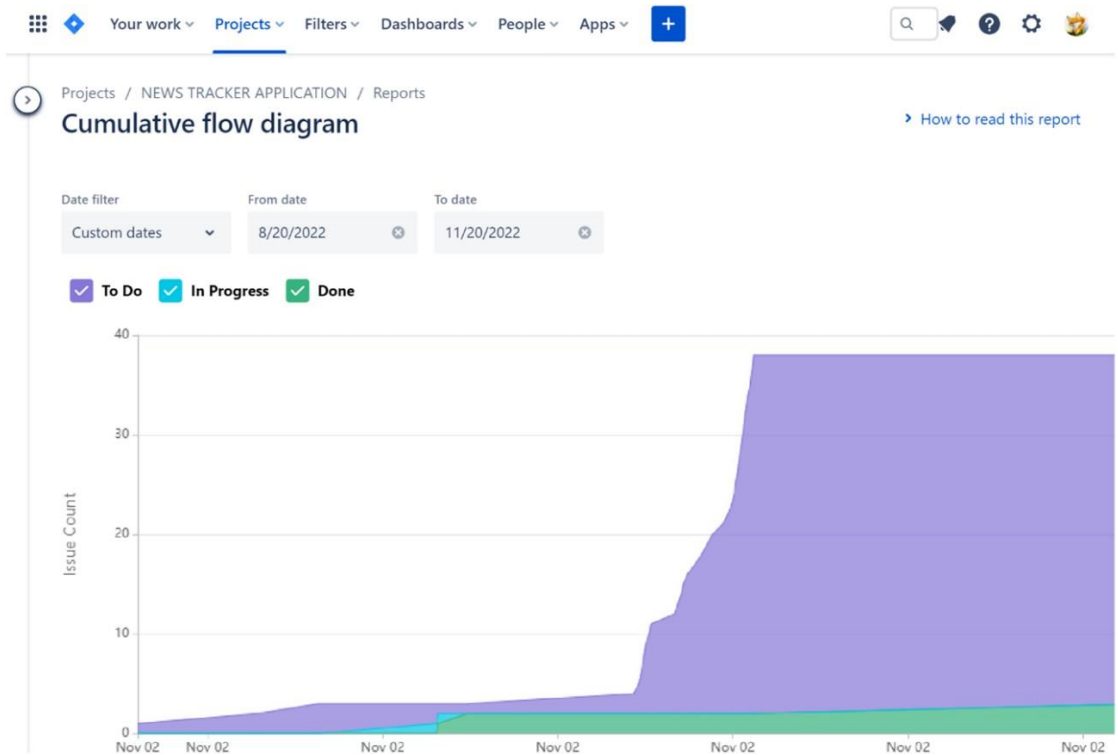
## Burndown Chart:



## Burnup report:



### Cumulative Flow Diagram:



## 7. CODING & SOLUTIONING

### 7.1 Real time news display

The application updates the news regularly i.e., whenever any new news gets posted this automatically updates in the application. This is achieved with the help of news API. Thus, whenever some news gets released, it is automatically updated in the app without any need human intervention.

```
newsapi = NewsApiClient(api_key='70fdb9ba81ba40b6bda148e672898bd9')

def get_sources_and_domains():
    all_sources = newsapi.get_sources()['sources']
    sources = []
    domains = []
    for e in all_sources:
        id = e['id']
        domain = e['url'].replace("http://", "")
        domain = domain.replace("https://", "")
        domain = domain.replace("www.", "")
        slash = domain.find('/')
        if slash != -1:
            domain = domain[:slash]
        sources.append(id)
        domains.append(domain)
    sources = ", ".join(sources)
    domains = ", ".join(domains)
    return sources, domains
```

## 7.2 Chat bot integration

The application has a chatbot integrated with it to help with basic user queries and to interact with the user. The chatbot feature is added to the application by using IBM Watson assistant in IBM cloud. This chatbot can interact with the user and guide them for simple queries.

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "2883db6a-d255-4bcc-b120-fde3b220264b", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "77e63bea-ee75-4371-abe7-ff2b31279d73", // The ID of your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion ||
    'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>
</body>

</html>
```

## 7.3 News search feature

The application is provided with a feature where the user can search the news that he/she wants using the search tool. This makes the app user friendly and provides the user to search directly for some news instead of scrolling and looking for a news.

```
class SearchCommand(Command, SessionCommandMixin):
    """Search for PyPI packages whose name or summary contains <query>."""

    usage = """
    %prog [options] <query>"""
    ignore_require_venv = True

    def add_options(self) -> None:
        self.cmd_opts.add_option(
            "-i",
            "--index",
            dest="index",
            metavar="URL",
            default=PyPI.pypi_url,
            help="Base URL of Python Package Index (default %default)",
        )

        self.parser.insert_option_group(0, self.cmd_opts)
```

## 8. TESTING

### 8.1 Test Cases

- Verify if user can login.
- Verify if the UI elements are getting displayed properly.
- Verify if the user can select the news displayed.
- Verify if the user can register.
- Verify if the application displays correct news.
- Verify if the user can logout.
- Verify if the user is getting redirected to the destination.
- Verify if the user can use the chatbot.
- Verify if the user login details are stored in the database.

### 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

| Resolution     | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|----------------|------------|------------|------------|------------|----------|
| By Design      | 3          | 2          | 2          | 1          | 8        |
| Duplicate      | 1          | 0          | 2          | 0          | 3        |
| External       | 2          | 0          | 0          | 1          | 3        |
| Fixed          | 6          | 3          | 4          | 6          | 19       |
| Not Reproduced | 0          | 0          | 1          | 0          | 1        |
| Skipped        | 0          | 0          | 0          | 1          | 1        |
| Won't Fix      | 0          | 3          | 1          | 1          | 5        |
| Totals         | 12         | 8          | 10         | 10         | 40       |

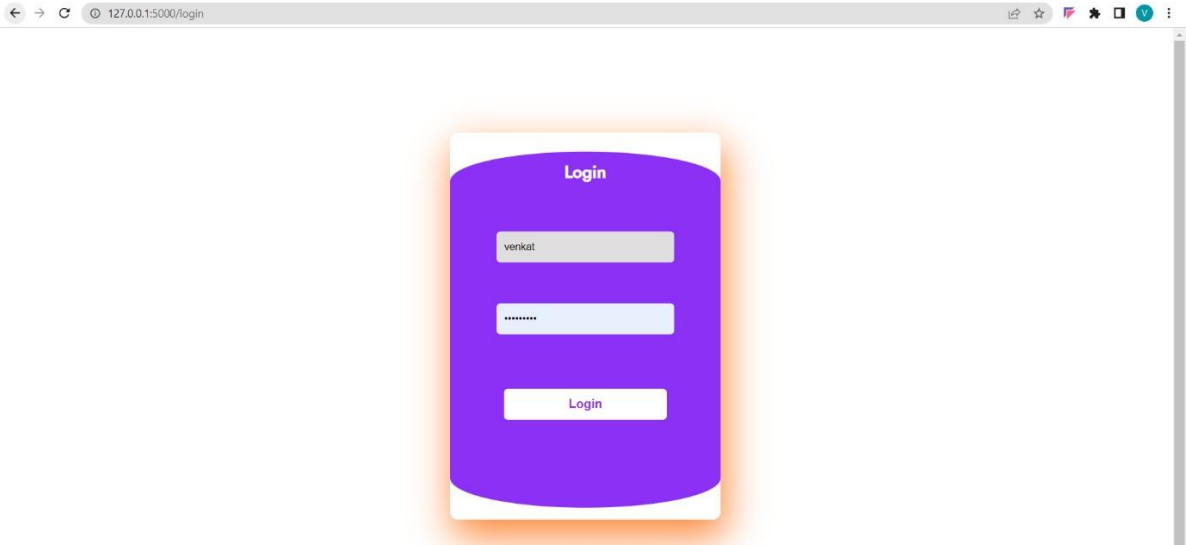


## Test Case Analysis

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

| Section             | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Print Engine        | 6           | 0          | 0    | 6    |
| Client Application  | 5           | 0          | 0    | 5    |
| Security            | 2           | 0          | 0    | 2    |
| Outsource Shipping  | 7           | 0          | 0    | 7    |
| Exception Reporting | 5           | 0          | 0    | 5    |
| Final Report Output | 4           | 0          | 0    | 4    |
| Version Control     | 2           | 0          | 0    | 2    |

# 9. Results



## Welcome To NewsApp

### Today's Headlines



Beer Sales Banned Around Qatar World Cup Stadiums: FIFA - NDTV Sports

According to the report, an unnamed official, who is aware of the chance of plans, has informed



Delhi Murder Case Updates: Court Orders Forensic Lab to Conduct Accused Aaftab's Narco Test Within Days - News18

## Welcome To NewsApp

### Today's Headlines



Beer Sales Banned Around Qatar World Cup Stadiums: FIFA - NDTV Sports



Delhi Murder Case Update! Lab to Conduct Accused Aaftab's Narco Test Within Days

Watson Assistant

Example: Type "hello"

Welcome to this News Tracker Application!!!  
Please Enter Your Email ID:

venkat01@gmail.com

Thank You For Sharing Your Email ID.  
Choose the type of News you wanted to read

International

International

For International News, Visit the below link:  
<http://127.0.0.1:5000/>

Type something

Activate Windows

Built with IBM Watson

## **10. Advantages and Disadvantages**

### **Advantages:**

- Easy user interaction.
- Faster loading of news.
- Easy login process.
- Real time news update.
- Chatbot integration.

### **Disadvantages:**

- Cannot view videos.
- Interface can be choppy at times.
- No advance security layer for data protection.
- No categories section.
- Algorithm may be slow at times.

## **11. Conclusion**

NEWS is the main source where people get to know about what's happening around the world. Reading news through devices like mobile phones, laptops has been widely adopted now. So, it's becoming necessary for an application to provide legit news. An application that tracks and shows user specific news has been developed as a part of our project. The app has been build using Flask and HTML, CSS, JS for the front-end. This app tracks news based on user preference and delivers the news to the end users. This app provides a more reliable and user-friendly way to read news.

## **12. Future Scope**

The following features can be added in the application in the future:

- To add feature to watch videos.
- To make the interface smoother.
- To add advance security layer for data protection.
- To add categories section.
- To update the app with a faster algorithm without affecting the advantages.

## 13. Appendix

### Source Code(app.py):

```
app.py U X
app.py > ...
1 import os
2 from flask import Flask, render_template, request, redirect, url_for, session
3 from newsapi import NewsApiClient
4 from flask_mail import Mail, Message
5 import ibm_db
6 import re
7 # init flask app
8 app = Flask(__name__)
9 app.secret_key = 'a'
10 conn= ibm_db.connect(("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SEC
11
12
13 newsapi = NewsApiClient(api_key='70fdb9ba81ba40b6bda148e672898bd9')
14
15
16 def get_sources_and_domains():
17     all_sources = newsapi.get_sources()['sources']
18     sources = []
19     domains = []
20     for e in all_sources:
21         id = e['id']
22         domain = e['url'].replace("http://", "")
23         domain = domain.replace("https://", "")
24         domain = domain.replace("www.", "")
25         slash = domain.find('/')
26         if slash != -1:
27             domain = domain[:slash]
28         sources.append(id)
29         domains.append(domain)
30     sources = ", ".join(sources)
31     domains = ", ".join(domains)
32     return sources, domains
33
34 @app.route('/login', methods = ['GET', 'POST'])
35 def login():
36     global userid
37     msg = ''
```

```
app.py > ...
```

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

```
def login():
```

```
36 |     global userid
```

```
    msg = ''
```

```
11 |     repeat, method = request.args.get('method', 'GET')
```

```
    username = request.args.get('username')
```

```
    password = request.args.get('password')
```

```
    sql = "SELECT * FROM users WHERE username =? AND password=?"
```

```
    stmt = db.prepare(conn, sql)
```

```
    db.cursor().bindparam(stmt, 1, username)
```

```
    db.cursor().bindparam(stmt, 2, password)
```

```
47 |     db.execute(stmt)
```

```
    account = db.cursor().fetchone()
```

```
49 |     print (account)
```

```
    if account:
```

```
51 |         session['loggedin'] = True
```

```
        session['id'] = account['USERID']
```

```
        userid = account['USERID']
```

```
        session['username'] = account['username']
```

```
        msg = 'logged successfully!'
```

```
        msg = 'logged successfully!'
```

```
        return redirect(url_for('home'))
```

```
59 |     else:
```

```
        msg = 'Incorrect username / password!'
```

```
    return render_template('login.html', msg=msg)
```

```
$app.route('/register', methods = ['GET', 'POST'])
```

```
def register():
```

```
    msg = ''
```

```
11 |     repeat, method = request.args.get('method', 'GET')
```

```
    username = request.args.get('username')
```

```
    email = request.args.get('email')
```



e nn arre = , fontt [ ' i r r a d ' ]

pa = , fontt [ ' n e s ' , and ' ]

i \* n d o b ind panarr stot, l, u = n o t )

ac = T b i d d h . let ch a s soc

, u = nn aoe )

i em U \* . bi nd pa ram map , l , u )

i em U \* . bi nd pa ram map , z , i 1 )

i em U \* . bi nd pa ram map , , pa word )

```

app.py > ...
100     mail = Mail(app)
101     recipient = request.form['email']
102     msg = Message('Successfully Registered', recipients=[recipient])
103     msg.body = ('Congratulations! You have successfully registered with '
104               'NewsApp!')
105     msg.html = ('<h1>Successfully Registered</h1>'
106               '<p>Congratulations! You have successfully registered with '
107               '<b>NewsApp</b>!</p>')
108     mail.send(msg)
109     return redirect(url_for('login'))
110 elif request.method == 'POST':
111     msg = 'Please fill out the form !'
112     return render_template('login-register.html', msg = msg)
113
114
115 @app.route("/", methods=['GET', 'POST'])
116 def home():
117     if request.method == "POST":
118         sources, domains = get_sources_and_domains()
119         keyword = request.form["keyword"]
120         related_news = newsapi.get_everything(q=keyword,
121                                             sources=sources,
122                                             domains=domains,
123                                             language='en',
124                                             sort_by='relevancy')
125         no_of_articles = related_news['totalResults']
126         if no_of_articles > 100:
127             no_of_articles = 100
128         all_articles = newsapi.get_everything(q=keyword,
129                                             sources=sources,
130                                             domains=domains,
131                                             language='en',
132                                             sort_by='relevancy',
133                                             page_size = no_of_articles)['articles']
134         return render_template("home.html", all_articles = all_articles,
135                               keyword=keyword)
136     else:
137         top_headlines = newsapi.get_top_headlines(country="in", language="en")

```

```

top_headlines = newsapi.get_top_headlines(country="D ", language='ea")
total_results = top_headlines['totalResults']
if total_results > 100:
    total_results = 100
all_headlines = newsapi.get_top_headlines(country="D ",
                                           language="en",
                                           page_size=total_results)
return render_template("home.html", all_headlines = all_headlines)

if __name__ == "__main__":
    app.run(debug = True)

```

**GitHub :**

**<https://github.com/IBM-EPBL/IBM-Project-50546-1660915638>**