

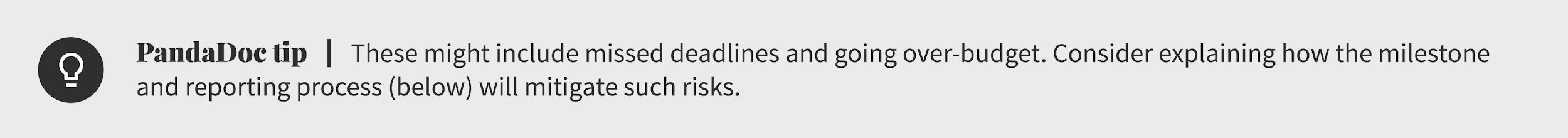
MEDIA BRAIN

INITIAL SOLUTION PROPOSAL

April 20, 2023

# 

# EXECUTIVE SUMMARY

The basis for this document is the design slides pdf and the discussion call held on April 06, 2023 for understanding the flow and business use case of the product. Since we have decided to develop only some features for the sake of POC, the architecture/solution would be implemented partially.

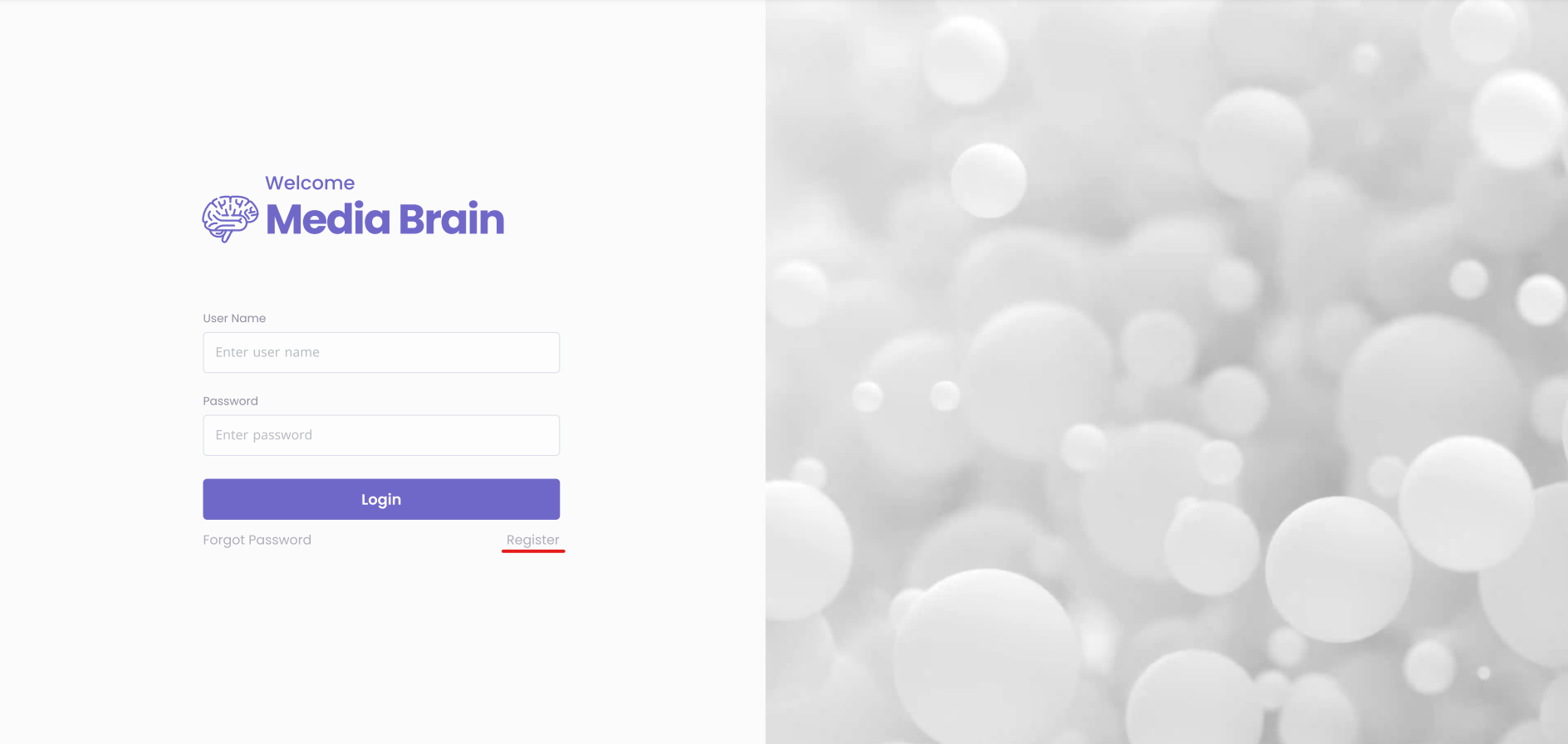
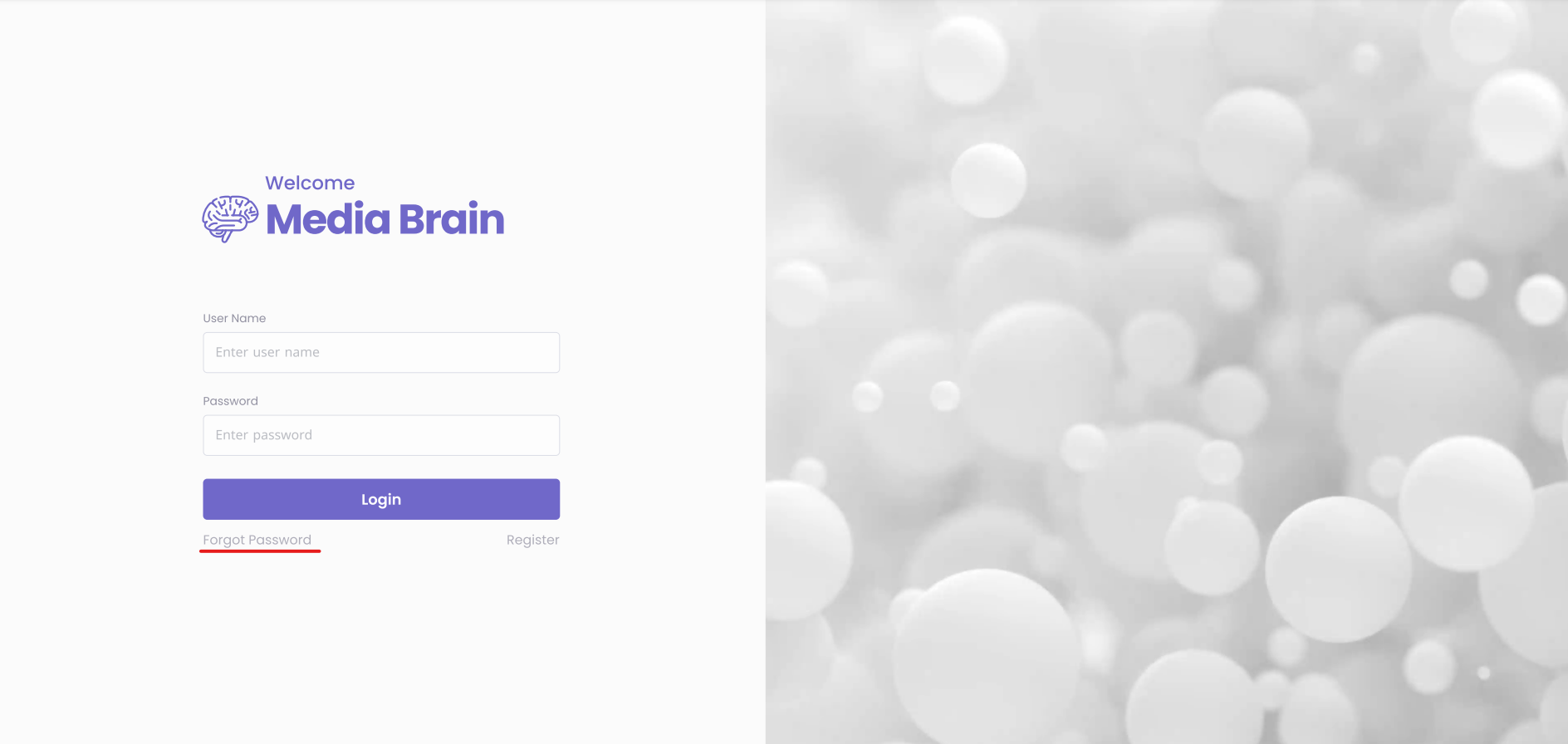
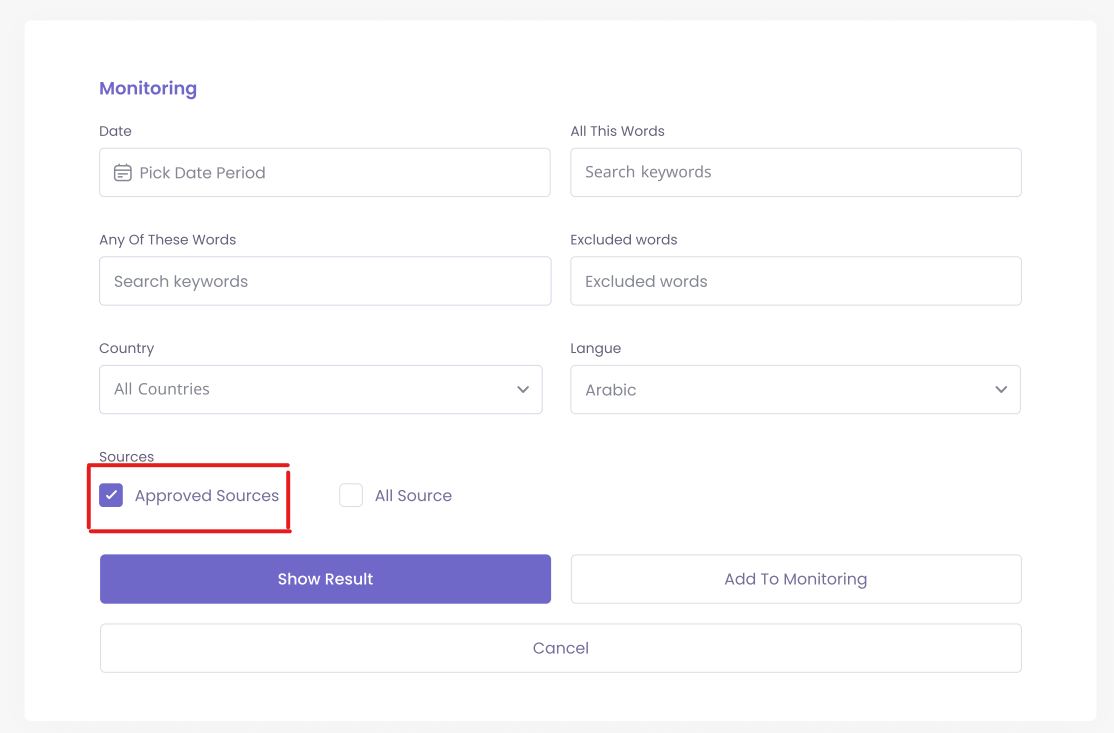
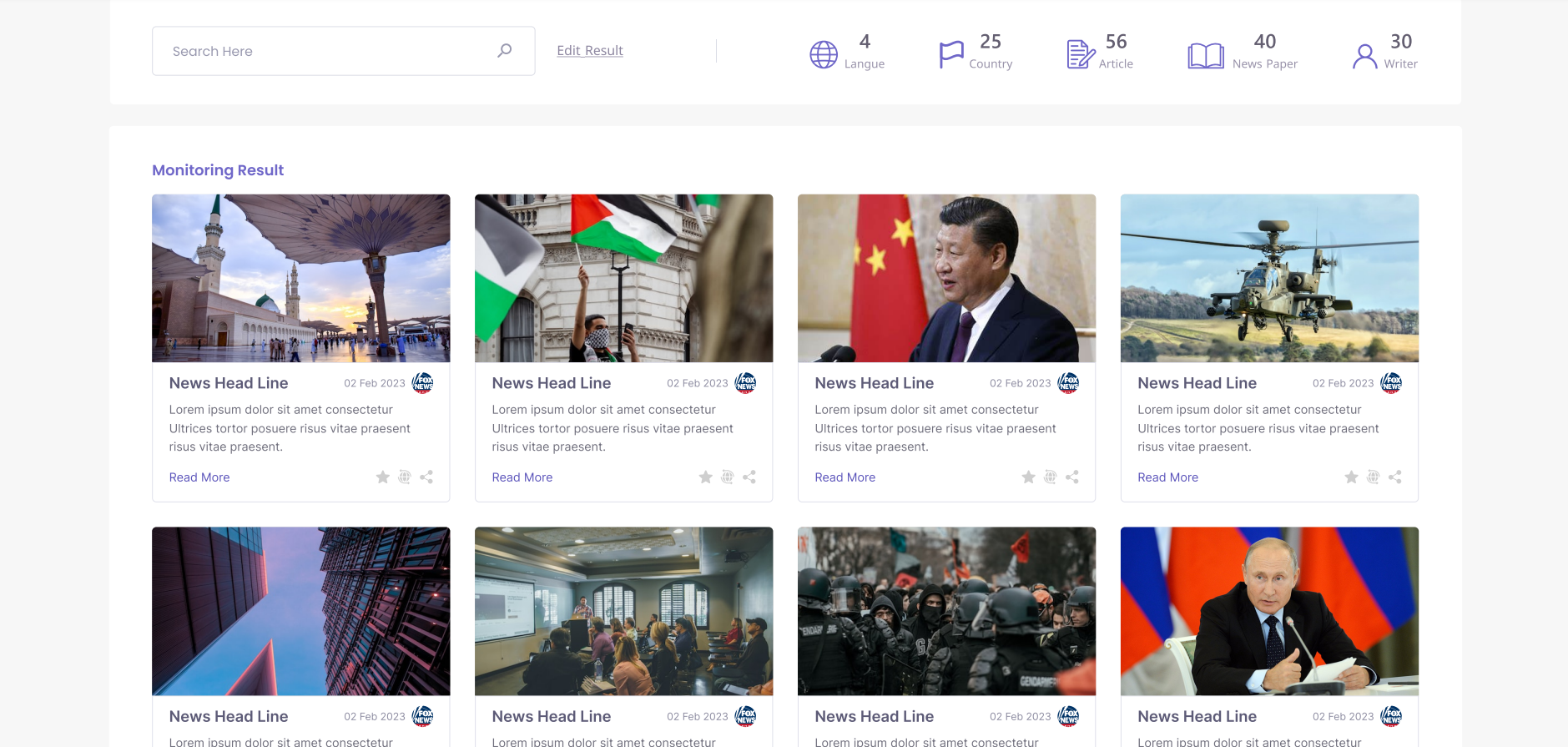
NOTE: The solution is subject to change depending upon extension or curtailment in the business use case or scalability issues.

## REQUIREMENT

This will include all the gathered technical requirements to simplify the development process, clarify the problem understanding and act as our foundation when creating our user stories for development.

* User login form
* Application translation (English - Arabic)
* News type selection
* News actions options (Monitoring, analytics, AI, etc.)
* Monitoring Input form
* Saving the monitoring input
* Viewing monitoring list
* Displaying monitoring results
* Analytics dashboard
* Artificial Intelligence dashboard
* Media indicator options
* Displaying writer indices list
* Displaying newspaper indices list

**There are some requirements that were not mentioned/displayed in the design slides pdf, but considering the whole application flow, we thought it would be required.**

* **Media-house registration form:**
  + In the login screen in the pdf, there's a register button. But, what screen would be getting displayed once clicked on that needs to be known for preparing the dataset for storing a customer’s information (in our case the media organizations).
* **Forgot password form and flow:**
  + Although in the login screen in the pdf, there’s a link/button of forgot password. But, what flow would be getting followed once a user clicks on it, needs to be discussed.  
    As per our assumption, the user would be required to enter his/her email address and then a tokenized link would be mailed to the user to reset his password.
* **User profile page:**
  + In the design slides, there’s no screen for viewing the logged in user’s information (personal information, organization information, etc).
* **Media-house users’ table:**
  + If in case the user has admin rights, then he could also be shown the list of invited members from his organization who are using the Media Brain platform.
* **Invite new member:**
  + The organization/media-house admin can invite his colleagues to use the Media Brain platform under their organization’s subscription.
* **Maintaining approved news sources as per the media organization:**
  + In the design slides, at slide no. 4, there’s a checkbox for getting the news from approved news sources. So, are we maintaining the approved news sources data from our own which would be common for all the users or the users have the ability to maintain their own list as per their preferences.
* **Viewing detailed news information of a particular monitoring result news:**
  + At slide 8, if the user clicks on any of the news results, then how would the detailed news be displayed to them. There’s no screen displaying the detailed news information.
  + 

**There are some requirements which we thought could be skipped in the POC. But, the final decision for these is awaited till the discussion.**

* News translation
* Displaying VIP indicator list
* Kingdom indicators dashboard
* Configuring notifications
* Sharing and marking the news as favorite

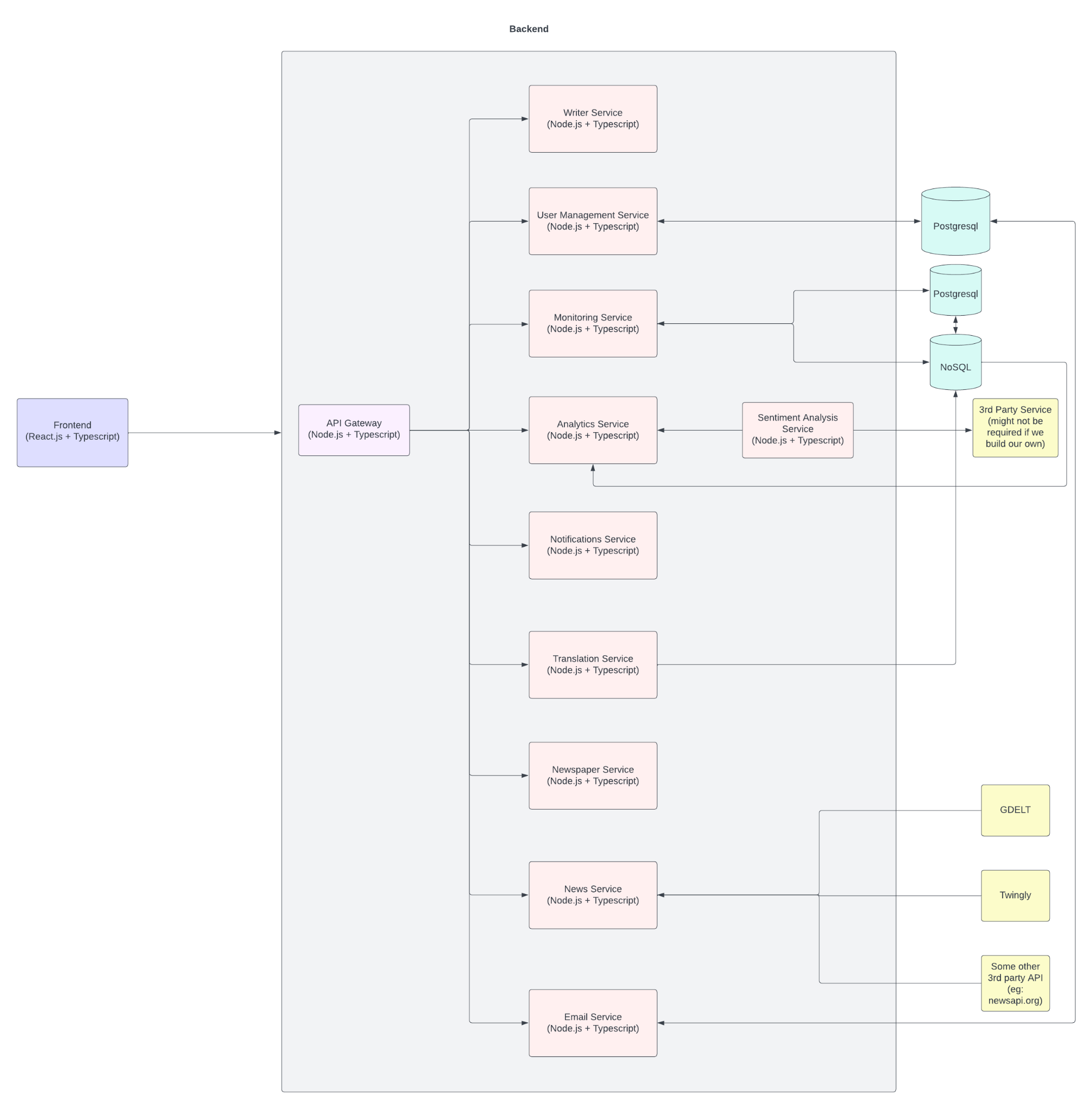
# TECHNICAL STACK

* **Frontend**
  + React.js
  + Typescript
  + Webpack
  + Babel
  + Material UI
  + Redux toolkit
* **Backend**
  + Node.js
  + Typescript
  + Graphql
  + REST
  + Apollo express server
  + Express.js
* **Database**
  + Google Big Query
  + Postgresql
* **Others**
  + Cron Job ( yet to be decided if its required)
  + Google Cloud Natural Language API

# SUGGESTED SOLUTION

## OVERVIEW

The current suggested design focuses on concern of separation and basically makes each integration as a separate service. Each building block can and will use the other blocks to achieve their own responsibility.



## SERVICES

1. **USER MANAGEMENT SERVICE**

This service would be responsible for handling all the application users’ data according to the roles assigned to them and would also manage all the authentication flows that are included within the business case of the application.

1. **NEWS SERVICE**This service would be built as per the adapter design pattern. The sole purpose of this application would be to fetch the news from various news sources (GDELT, Twingly, Newsapi, etc).
2. **MONITORING SERVICE**This service would be responsible for storing the monitoring choices made by the users into the database. This service would also consume [News Service](#deajehs3z3zo) to serve the monitoring results to the users as per their monitoring preferences. This service would also store the monitoring result into the database.
3. **ANALYSIS SERVICE**As the name suggests, this service would be responsible for performing analytical calculations over the monitored news data. It would provide all the calculated data to the frontend for generating the analysis charts for the users. For fast performance of the application, it would also store the analysis results into the database according to the monitoring results. So, if the user wants to see the analysis of the previous monitorings we don’t have to calculate it again.
4. **SENTIMENT ANALYSIS SERVICE**  
   The sole purpose of this service would be to do processing over the news results and prepare sentiment analysis data from that. We would be either using some 3rd party service (eg: Google Cloud Natural Language API) or would be writing our own AI/ML NLP algorithm for doing the sentiment analysis.
5. **EMAIL SERVICE**This service would be responsible for sending mails to the users wherever required in the application. It would be either getting used directly by the frontend or would be consumed by other API services for sending the mails with the help of Sendgrid.
6. **NEWSPAPER SERVICE**  
   This service is responsible for maintaining the data of the various newspapers. It would store all the required information regarding the newspapers and would serve that to the frontend or other API services whenever required.
7. **WRITER SERVICE**  
   As like the previously mentioned Newspaper service, this service would be responsible for maintaining the data of various authors and writers. It would store all the required information regarding the authors & writers and would serve that to the frontend or other API services whenever required.

## ASSUMPTIONS

If our news provider faces downtime for some reason, then in that case our application might also not work, as we are dependent on the news provider API to fetch us the data.

So, in this case we thought to store the data of every monitoring that the user does which is saved to the monitoring list. The data would be stored in some NoSQL database.

Then in the downtime time period we would disable new monitorings but the user can still view the results of the previous monitorings which he saved into the monitoring list via our saved data in the NoSQL database.

# MVP MILESTONES

## OVERVIEW

For getting continuous feedback and doing prompt work over those feedbacks, we have decided to break the overall POC into certain milestones. The deliveries would be given according to each milestone.   
Post each milestone, we would work over the feedback (if any) and then after finishing that, will continue our development for achieving the next milestone.

Hence, we have divided the whole project into certain milestones, with the list of features that would be delivered with each milestone.

## MILESTONE - 1

* Media house registration
* User login
* Displaying news options
* Displaying news actions options (Monitoring, analytics, etc.)
* Monitoring input form
* Save to monitoring list
* Displaying monitoring list

## MILESTONE - 2

* Forgot password
* User profile page
* Invite new user
* Edit/Update organization’s users details
* Add/delete approved news sources
* View monitoring results

## 3. MILESTONE - 3

* View detailed news from monitoring results
* Analytics dashboard
* Artificial Intelligence Dashboard
* Media indicator options
* Displaying writer indices list
* Displaying newspaper indices list

# TIME ESTIMATES

| **Frontend** | **Efforts (hrs)** |
| --- | --- |
|  |  |
| **SETUP** | |
|  |  |
| Repo Setup | 12 |
| MUI setup & Theme configuration | 16 |
| App LanguageTranslation setup | 16 |
| Layouts Designing | 24 |
| Redux & Routing setup | 16 |
|  |  |
| **COMPONENTS & PAGES** | |
|  |  |
| Generic Components | 80 |
| Chart components (7) | 70 |
| Map Component | 16 |
| Login Page | 16 |
| News Type Page | 8 |
| News Options Page | 8 |
| Monitoring Form | 16 |
| Monitoring List Page | 24 |
| Monitoring Results page | 24 |
| Analytics Page | 24 |
| Artificial Intelligence Page | 32 |
| Media Indicators Page | 4 |
| Writer Indices Table Page | 16 |
| News Paper Indices Table Page | 8 |
|  |  |
| **NOT MENTIONED IN PPT** | |
|  |  |
| User Profile page (User details and org users list) | 24 |
| Edit/Update Org User Details | 16 |
| Invite new org user | 16 |
| Add/edit approved news sources as per organization | 20 |
| Register Page | 16 |
| Forgot Password | 16 |
| Detailed view of news | 24 |
|  |  |
| **TOTAL** | **560** |

| **Backend** | **Efforts (hrs)** |
| --- | --- |
|  |  |
| **SETUP** | |
|  |  |
| Repositories setup | 12 |
| DB Design | 24 |
| Database setup & integration | 24 |
| Setup Google Big Query | 16 |
| Setup Google Cloud Natural Language API | 16 |
|  |  |
| **SERVICES** | |
|  |  |
| Newspaper Service | 24 |
| Writers Service | 24 |
| User Management Service | 56 |
| Monitoring Service | 80 |
| Analytics service | 80 |
| News Service | 48 |
| Email service | 24 |
| API Gateway | 32 |
| Auth and token verification | 24 |
| Sentiment Analysis Service | 40 |
|  |  |
| **TOTAL** | **528** |

| **DevOps / Deployment** | **Efforts (hrs)** |
| --- | --- |
|  |  |
| **TOTAL** | **160** |

| **Testing** | **Efforts (hrs)** |
| --- | --- |
|  |  |
| **TOTAL** | **300** |

**TOTAL EFFORT : 1550 hrs**