

# [PRD] Search Engine

**Summary:** Search engine for documents stored in the online cloud service

|  |  |
| --- | --- |
| **Created:** july 12, 2021  **Owner:** krishna.ramg3@gmail.com  **Contributors:** | **Status:** WIP | **In Review** | Approved | Obsolete  **RFC:** Link to RFC when created |



The Goal for this PRD to build a solution prototype for a search engine that can take a file from the cloud storage(read google) and make it searchable for the user.

## Background

Many people stored their documents in cloud storage like Gdrive, Dropbox, Box, etc.

Those data are not searchable at all. The only way you can look up a document is by filename

Or file metadata.

## Problem

Many people have a large number of text documents stored in cloud storage which contains a

Lots of sensitive and private information.

Indexing them to some public search engine will results in mainly leaking of private information, theft of identity, it also results in privacy issues.

## 

## Requirements and Phases

|  |  |
| --- | --- |
|  | **Requirements** |
| **Phase 1:** API for Downloading files  using google API | Must be easy for the user to just share the shareable link |
| API should be able to download file |
| **Phase 2:** API for searching Extracted data from the file | The downloaded file contents must be extracted and index into a search database and the search service should be accessible via a API |
| **Phase 3:** UI for API | UI where user can index their link using input box and search their data using the same |

### Phase 1:- Must be easy to share Google drive links

The user should interact with our system with ease

#### Share Url

The user will provide a shareable link of the document with our interface :

* API where the user can post the URL of the documents.
* API where the user can see the status of the file.
* use the APIs provided by the google online service to fetch the files stored in those services.

##### Acceptance Criteria

1. The file is downloaded into our system.
2. The user can see the status of the file whether it is downloaded or there is some error with the file

##### Considerations

1. Check if the file is already indexed in the system or not?

### Phase 2:- Search the Content

The file downloaded must be parsed and the extracted text content should be searchable using the API from the files

Should be indexed into ElasticSearch database service/other Search databases.

#### Requirement 1:- Extract the content

The file download data should be parsed to get contents. libraries like Apache Tika can be used to extract text from these documents.

The parsed data should be stored in ElasticSearch with file name, document URL, date, and extracted content

##### Acceptance Criteria

1. The user can see the status of the file whether data is extracted or not

##### Considerations

1. Check if the file is already indexed in the system or not?

#### Requirement 2:-

REST API takes the search term/token as input and returns a list of files and their HTTP URLs that contain the term.

a. E.g. curl -X GET https://<search-service-host>/search?q=”<PAN-NUMBER>”

##### Acceptance Criteria

1.Return a list of files and their HTTP URLs

##### Considerations

1. In case of no result what should be the response?

### Phase 3:- UI

A simple User Interface should be present where the user can input his Document url and should be able to search it.

#### Requirement 1:- Home Page

An HTML page where the user can input his Document URL and search his documents.

#### Requirement 2:- Search results

An HTML page where search results should be shown to the user.

## Approvals