**FINAL PROJECT REPORT**

**OPEN-DATA SEARCH**

**PROJECT DEFINITION:**

***Open-Data Search*** is a web application, a go-to place for searching public datasets on the Internet.

**PROJECT SCOPE:**

We have curated a database of public data from the website: UCI ML Repository making it easily searchable through our application.

It should be noted that we only maintain metadata of these datasets like dataset description, feature information, link to the dataset etc. (in other words, we have an index of datasets publicly available on the internet*,* not a database of datasets itself)*.*

The idea is to make the database crowdsourced and community maintained.

* User Signup/Authentication.
* User on logging in, will be able to add their own datasets (hence crowdsourced),
* User can up-vote/down-vote on existing datasets through Facebook, which will be displayed on the application also.

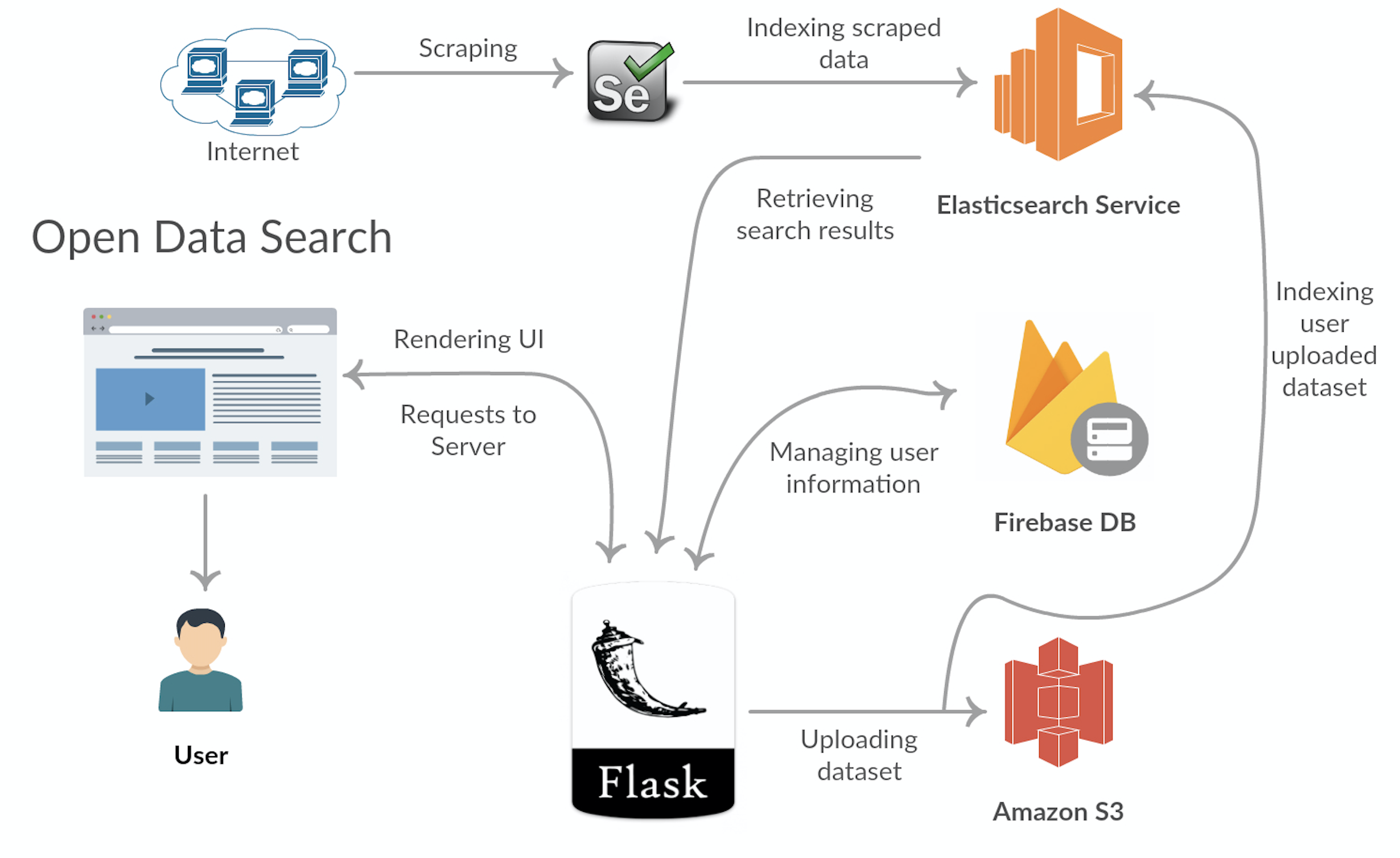
**FUTURE SCOPE:**

* Adding dataset recommendation system to the application, which could be either just content-based recommendation or collaborative recommendation from similarity of user behavior.
* Including data from other sources (which is only additional scraping effort).

**TECHNOLOGIES:**

* HTML, CSS
* JavaScript
* Google Firebase (User Management)
* Python Flask (Server)
* Elastic Search (Search Index of dataset information. – AWS Elastic search service)
* Amazon S3 to store user uploaded datasets.

**ARCHITECTURE:**



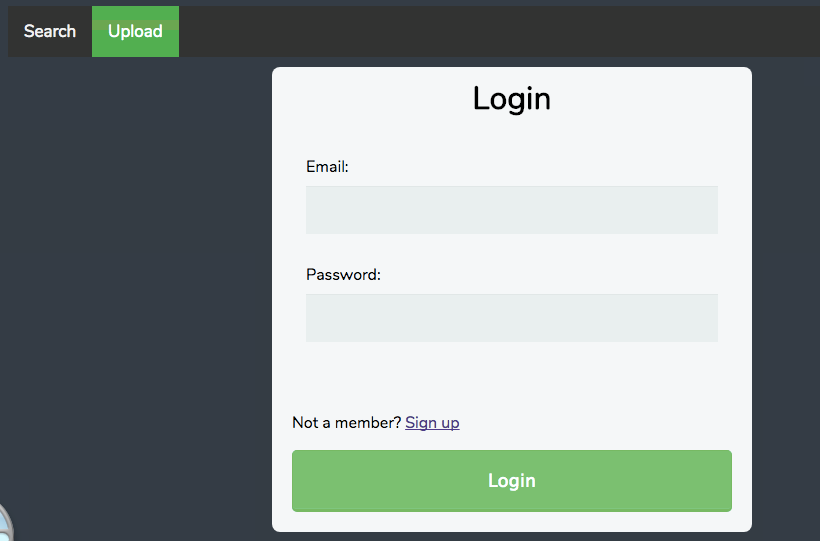
**PROGRESS:**

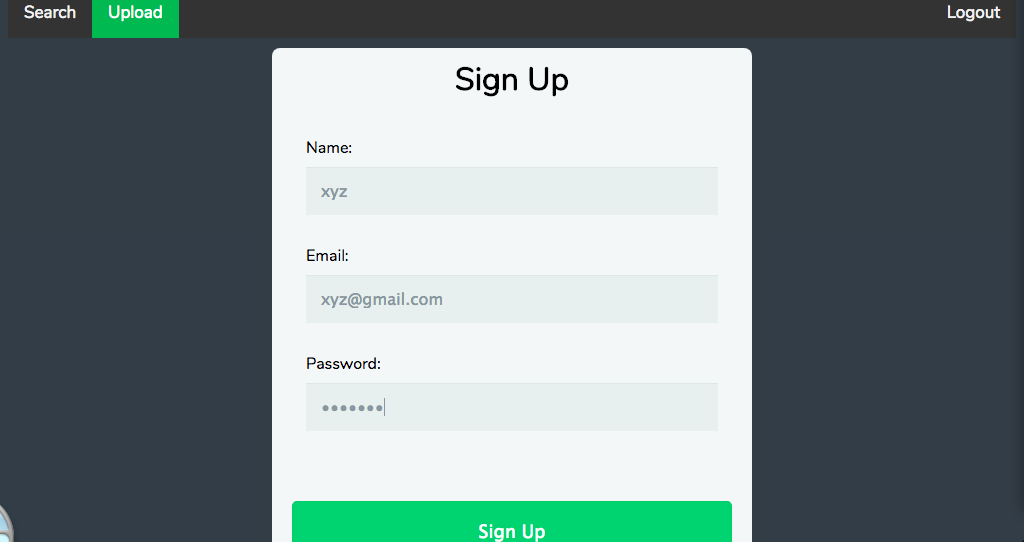
The completed components of the project are described below:

**SCREENSHOTS:**

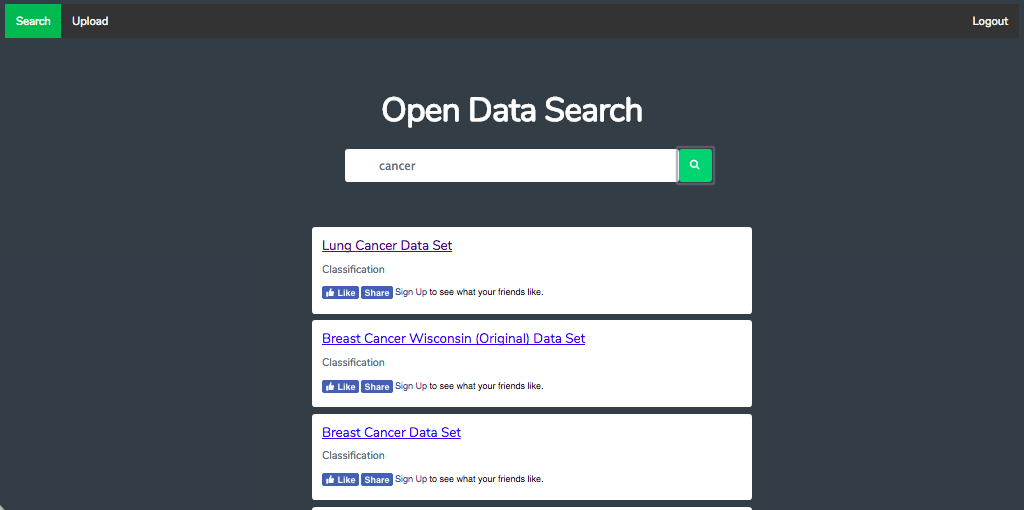
User login/Authentication:

Users can login/signup with their email and password. The signup/login page looks as below:



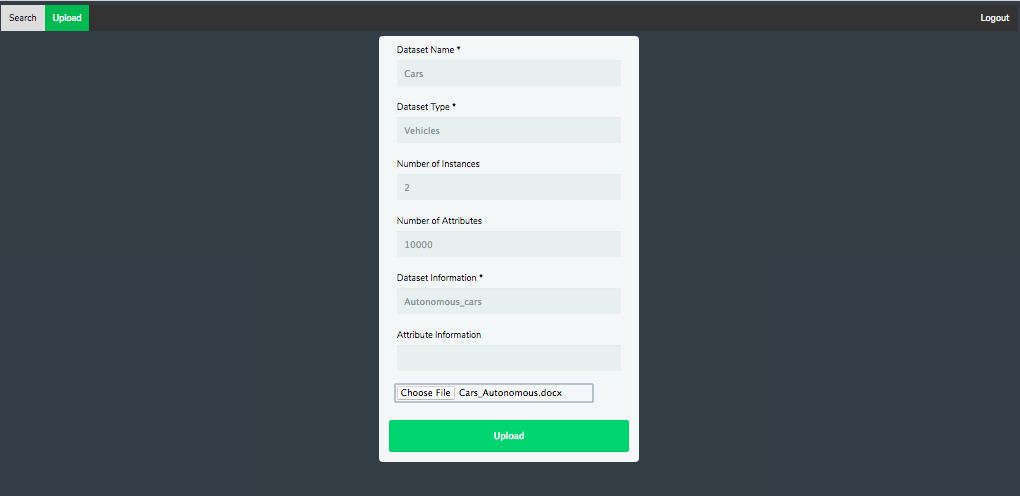


Our web application displays all the relevant datasets based on the search keyword. The hyperlink of the dataset along with the type of the dataset is displayed. For instance, in the first case, the datasets to be searched is displayed in the Search box as shown below:



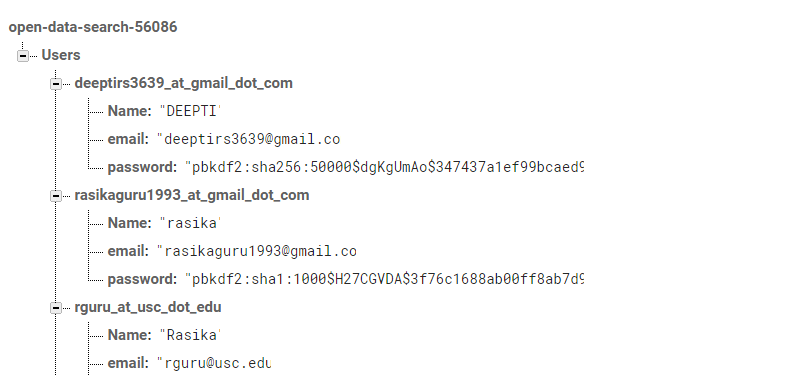
Here a normal string ‘Cancer’ is searched and all the datasets that contains the search query is retrieved. A ‘multi-match’ query is made in elastic search, which searches the query text in multiple fields (Name, Description etc.).

Users can upload new datasets with the relevant information like dataset Name/type, number of attributes, etc as shown below:



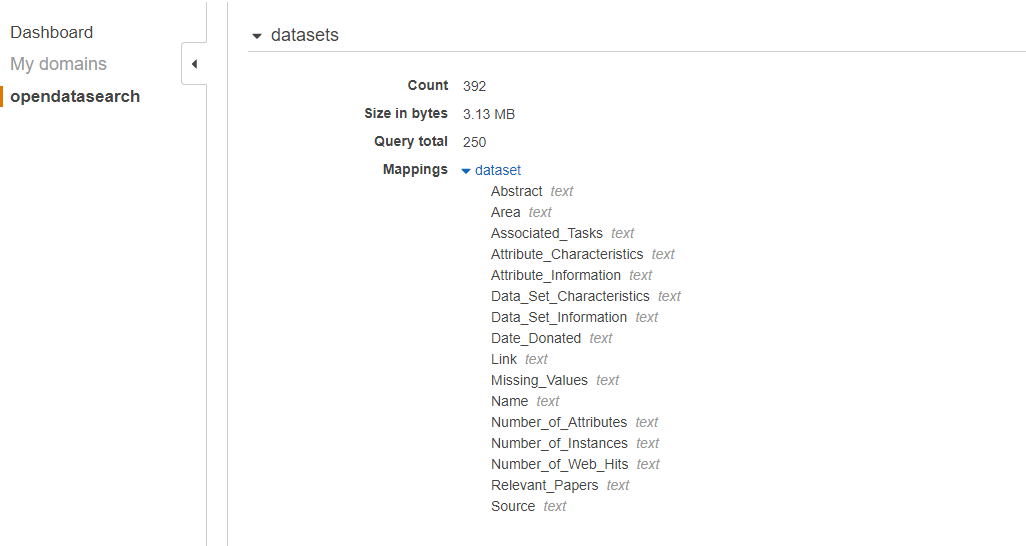
**FIREBASE:**

User information during authentication is stored in firebase. It has emailed, user name and the sign in password for every user. The password from the user is encrypted and stored in the database.



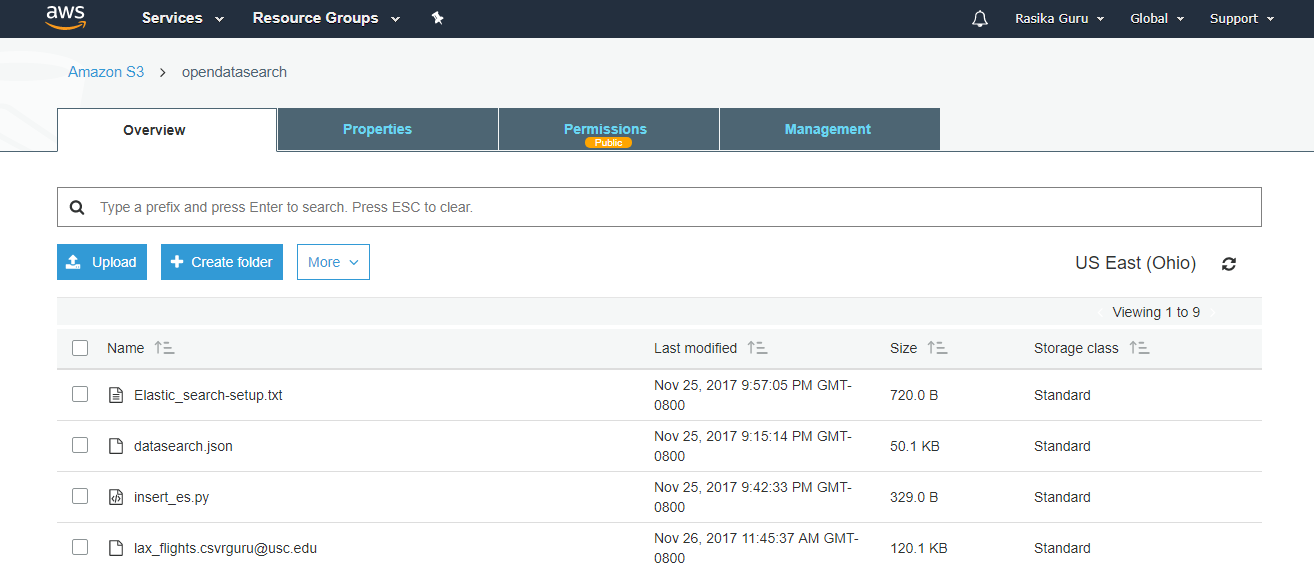
**AWS ELASTIC SEARCH:**

Contains all the scraped metadata information from UCI Machine Learning Repository for all the 392 datasets.



**AWS S3:**

AWS S3 holds the user uploaded datasets, and the corresponding information about the datasets from the user is indexed into Elastic search



**Social Plugin (with Facebook):**

The feature of liking and sharing the search results were implemented using the Facebook Social Plugin through a developer account. Liking the dataset on the website records that in your Facebook activity log and sharing it will share the dataset URL on your Facebook timeline. The likes are also displayed on the application.

**CHALLENGES:**

* Web Scraping was a challenge because of the absence of identifiers to locate a web element uniquely in the web page that was scraped (Unique class names/ids)
* The app was initially entirely client-side - hosted with firebase and search feature was implemented using the jQuery browser build. This had lack of flexibility in implementing other features like login/upload. So the app had to be migrated to Python Flask backend from firebase backend.

**TIMELINE:**

**GROUP**

Team members and Responsibilities are as below:

* Rasika Guru : UI Design, Elastic Search & S3 integration, Server-side Flask implementation
* Deepti Rajashekharaiah Siddagangappa: Elastic Search to retrieve data.
* Kavya Sethuraman: Web Scraping for dataset information.

|  |  |  |
| --- | --- | --- |
| **NAME** | **USC ID** | **EMAIL ID** |
| Rasika Guru | 4615679320 | rguru@usc.edu |
| Kavya Sethuraman | 7852999061 | ksethura@usc.edu |
| Deepti Rajashekharaiah Siddagangappa | 1716300496 | drajashe@usc.edu |