Docker

Document Search Engine

Introduction :

Document search engine where students can find required document. This is clone of Google search engine where student finds documents with filtering and based on search queries.

Purpose :

Document Search engine is a web platform which can be used by Institute and organizations where people can upload documents and when required people can search for documents by applying filtering ( like author , tags, downloads , dates uploaded) and download it.

Scopes

* Teachers can upload documents on the application
* Students can search required documents by filters and download it.
* Organization can use analytics tool.

Tools Used

* Django 1.9 Python Framework
* Django Rest Framework API
* AngularJs
* Bootstrap 1.8
* HTML 5 , CSS , JavaScript

References

* Django Documentation - <https://docs.djangoproject.com/>
* Django REST framework – [www.django-rest-framework.org/](http://www.django-rest-framework.org/)
* Bootstrap - <http://getbootstrap.com/>

Technologies to be used

* Relational Database Management System
* REST API

Software Interface :

- The web is used to interface.

- Web Browser ( Chrome, Firefox , Opera , Safari )

Hardware Requirements :

* No special Hardware requirements.

Performance Requirements :

* A Powerful Server . Data accesses is via API calls that leads to load on server.
* Good Hard Disk to store documents. Fast and Big.

API :

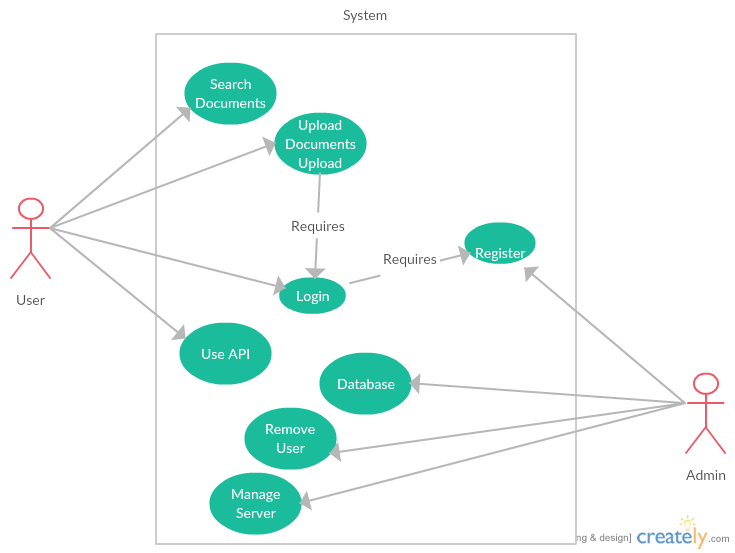
host : [http://127.0.0.1:8000](http://127.0.0.1:8000/)­

* host/api/users
  + Mehtod : GET
  + Gives list of all users

* host/api/users/{user\_name}
  + Method : GET
  + Gives User detail of {user\_name}
* host/api/documents
  + Method : GET
  + Gives list of all documents
* host/api/documents/{doc\_id}
  + Method : GET
    - Gives detail of document having doc\_id
  + Method : DELETE
    - Deletes document having doc\_id
  + Method : PUT
    - Updates entries of document having doc\_id
    - Send a form along with HTTP request using POSTand file in request BODY

* host/api/documents/upload
  + Uploads new document
  + Send data along with POST and file in request BODY
* host/api/search
  + Search document with specific constraints
  + Send json data with following keys [ can use whichever is required ]
  + doc\_uploaded\_by , doc\_title , doc\_tags, doc\_description , doc\_type

Use-case Reports



Class Diagram :

