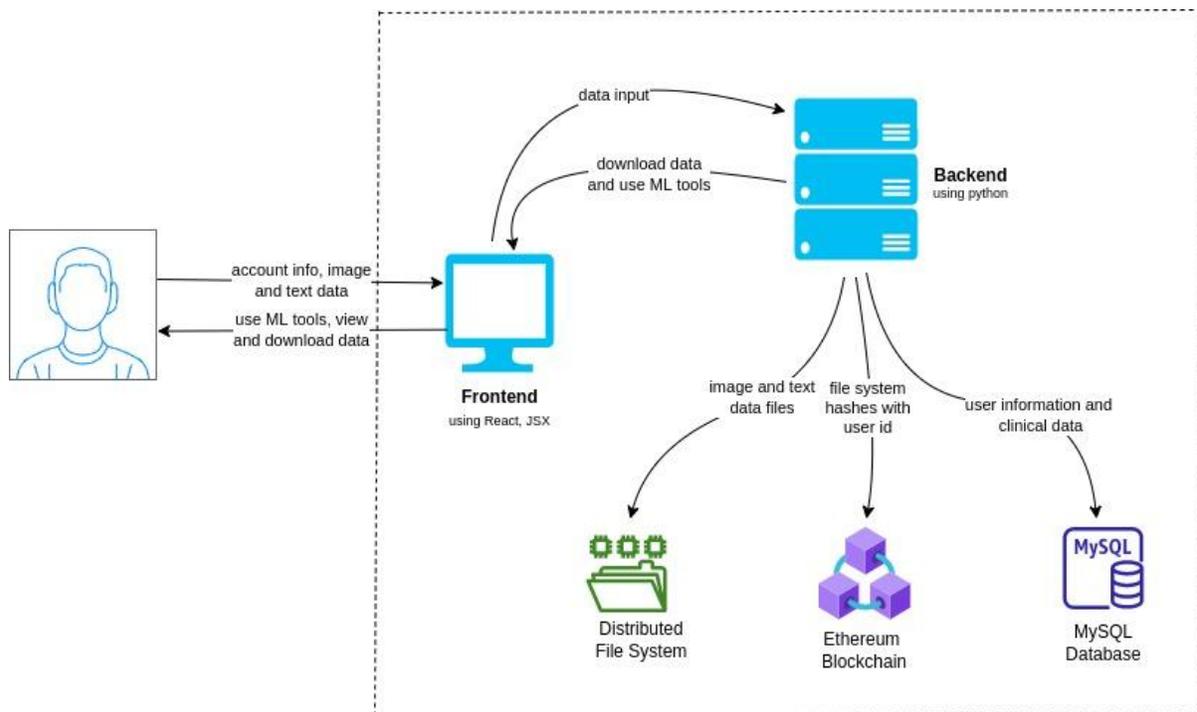


System Structure and Functioning

Structure :

The system is based on following general architecture.



As shown above, the system is divided into two distinct parts, frontend and backend. Both the frontend and backend reside in the same physical machine. To sustain the system, both frontend and backend reside in two different containers which are virtualized using Docker. The backend provides localhost based APIs which are used by the frontend to get and fetch resources. This system uses blockchain which needs at least two different servers or virtual server instances. If the system scales up, then load balancers will be needed for smoother operation. All the servers need 24x7 uninterrupted standard power supply. The server that mine the blockchain needs to have a good cooling system as the mining operation is resource consuming.

For the operation of the application a person with minimal knowledge of computers is sufficient. For data input, a user just needs to know how to fill up a form and upload files from his local storage system. For viewing and downloading data, the user just requires the knowledge

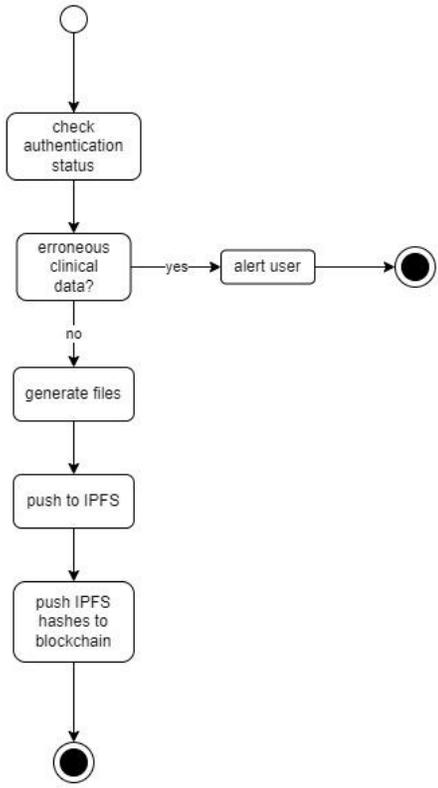
of browsing using a browser like Chrome. To use the machine learning tools in the system, user should possess enough knowledge of machine learning algorithms. Along with that, registering oneself with proper username and password and keeping them safe is another desired ability.

This application is good enough to fulfill the requirement of robust storage of medical images with textual clinical data while keeping the private patient information confidential. Since the storage is decentralized it is quite safe and immune of any cyber attack. To run the application, at least three instances of the server need to be active all the time which needs standard power supply in uninterrupt manner. Hence the system designed is economically quite efficient. Also that the users need minimal skill to operate makes it practically useful.

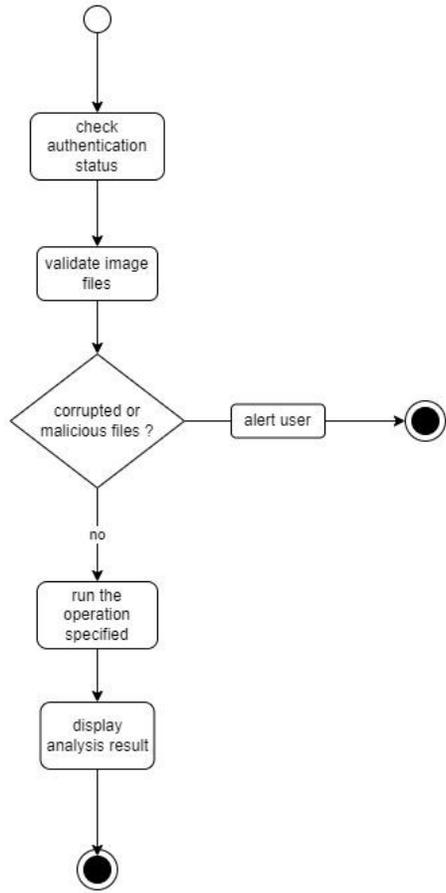
System Functions

The activity sequence that needs to be followed is explained here. A sidebar shows all the options to navigate around this application. A user can view available resource and sample images without the need to log in or register. To download full dataset or patient-wise dataset the user needs to either log in or register to create an account. After logging in, they can go for downloading the aforementioned. For uploading resource, a user must be logged in and given appropriate privilege level. They can decide to upload a full set of data i.e text and image data or text only or image only data (for uploading image only dataset, they must be aware of the patient's actual medical-id). To use the machine learning and preprocessing tools, user needs to log in first and then upload the target image and process the image. The output is displayed side by side of the original image uploaded by the user. A superuser, who have the highest privilege level also can change critical system features like encryption key or any key element of blockchain operation.

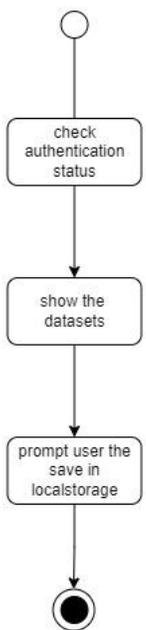
The above operations can be visualized in the following activity diagram.



Clinical Data Upload



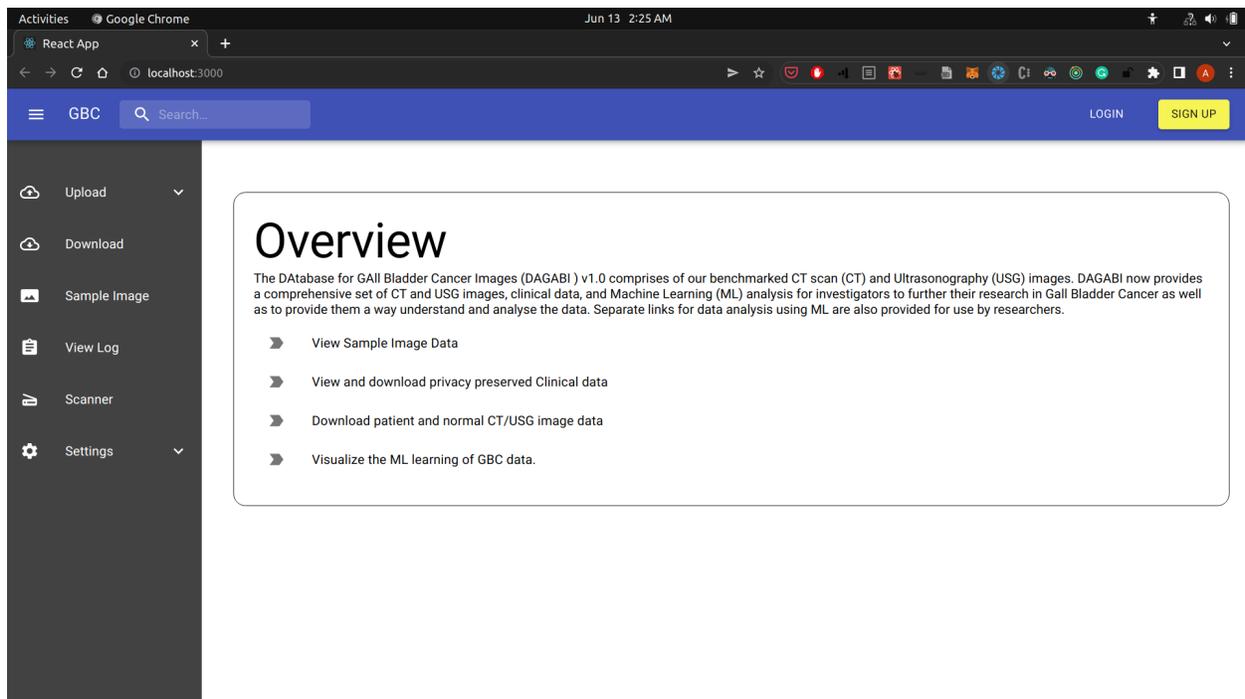
ML Tool Operation



Clinical Data Download

System Operation

A user needs to visit the [URL](#) to use the application. The landing page shows a brief introduction of the system and its purpose. This also contains a sidebar and navigation bar which contains tabs to different UI of operations. The navigation bar contains a login and registration button on clicking which renders UI for the same. The sidebar contains links to upload, download, sample image, system log, scanner and setting. This is as shown below.



Screenshot : Application Homepage

On clicking, Upload tab expands and shows three more options, upload text and image dataset together (full dataset), upload image only and text only (csv format). But if the user is not logged in then a login screen pops up. There are three modes of data upload, which are

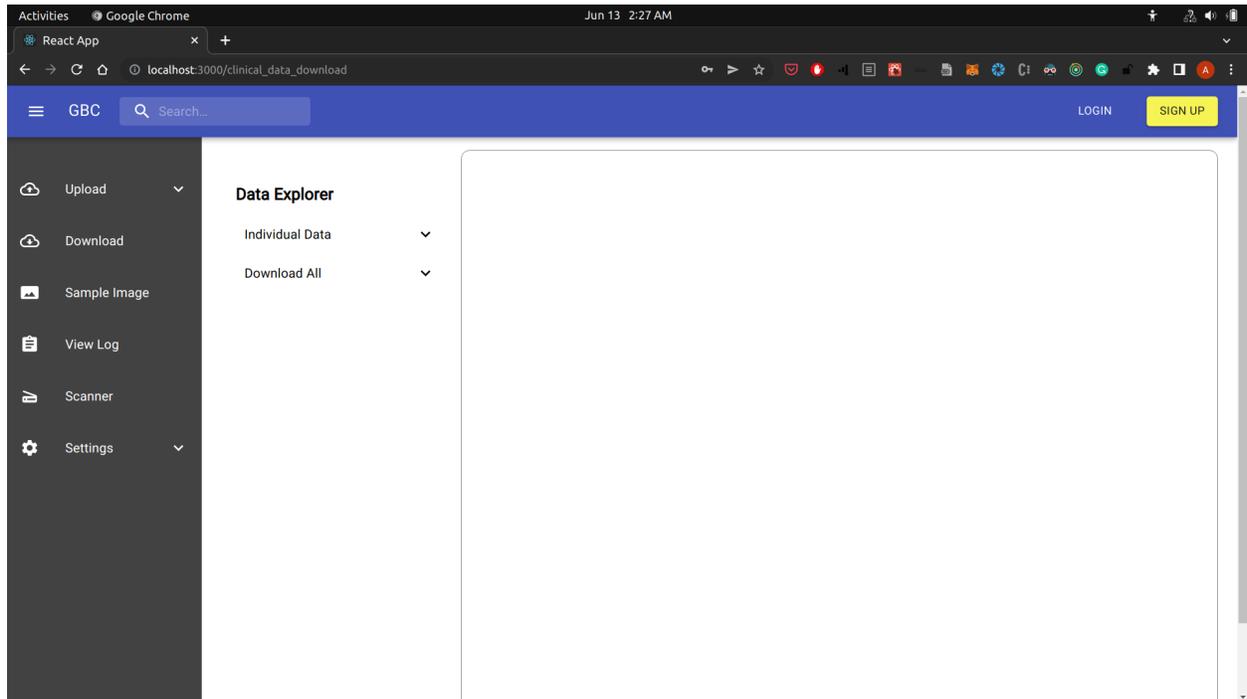
For full dataset upload, user needs to fill up a form where he needs to give details of a patient's medical condition. Then upload a zip file that contains dicom and nrrd files of the patient.

For uploading image only, the user needs to upload the zip file by choosing proper patient-id from a select menu.

For uploading text only data in csv format, the user needs to simply upload the csv file, records will be extracted in the backend automatically. But the user needs to make sure that csv file contains data in correct format (mentioned below).

In data download operation, by clicking the link in the sidebar for download user can open a page, where he can choose to download individual data or entire dataset altogether.

(need to take a screenshot showing multiple patients)



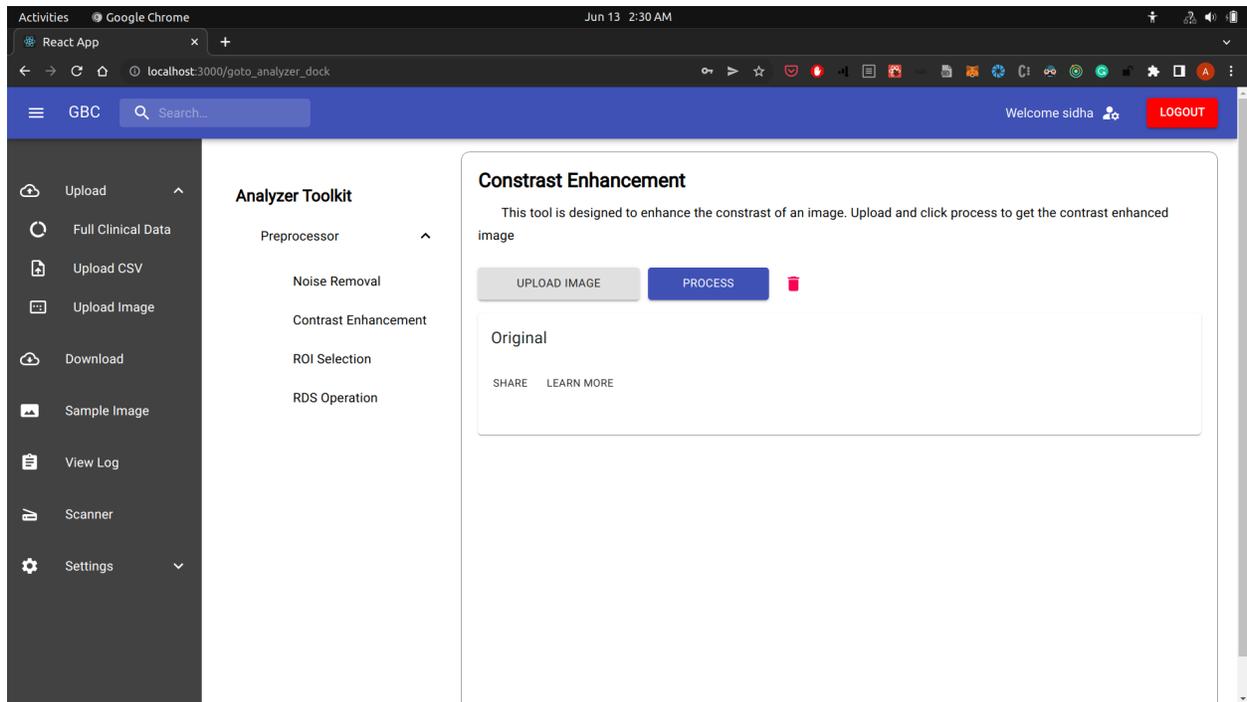
Screenshot : Data Download Page

For downloading data, the user needs to log in or register.

For viewing or downloading sample image, user needs to click the link mentioned as sample image. No login is required to use this facility.

Another important operation that is provided is to use machine learning and preprocessing tools. User again needs to log in to use this facility. On clicking the link mentioned as “Scanner”, the user choose from an array of tool set as to what operation to perform. Primarily there are two groups, “preprocessor” and “classifier”. Both contains the type of tools as indicated in the names. For performing any operation on an image, first he needs to upload the target image in a file input box, the page gives a preview of the image and two buttons either to remove the image or further process the image is shown. On clicking “process” button, the image is sent to backend and then processed image is shown side by side of the original image in the page. Similar method

or uploading and previewing image is implemented in all the tools. The following screenshot shows the page.



Screenshot : ML Tools

Other facilities of the system includes viewing the log records of activity, changing setting of the system which requires user to have the highest user privilege.