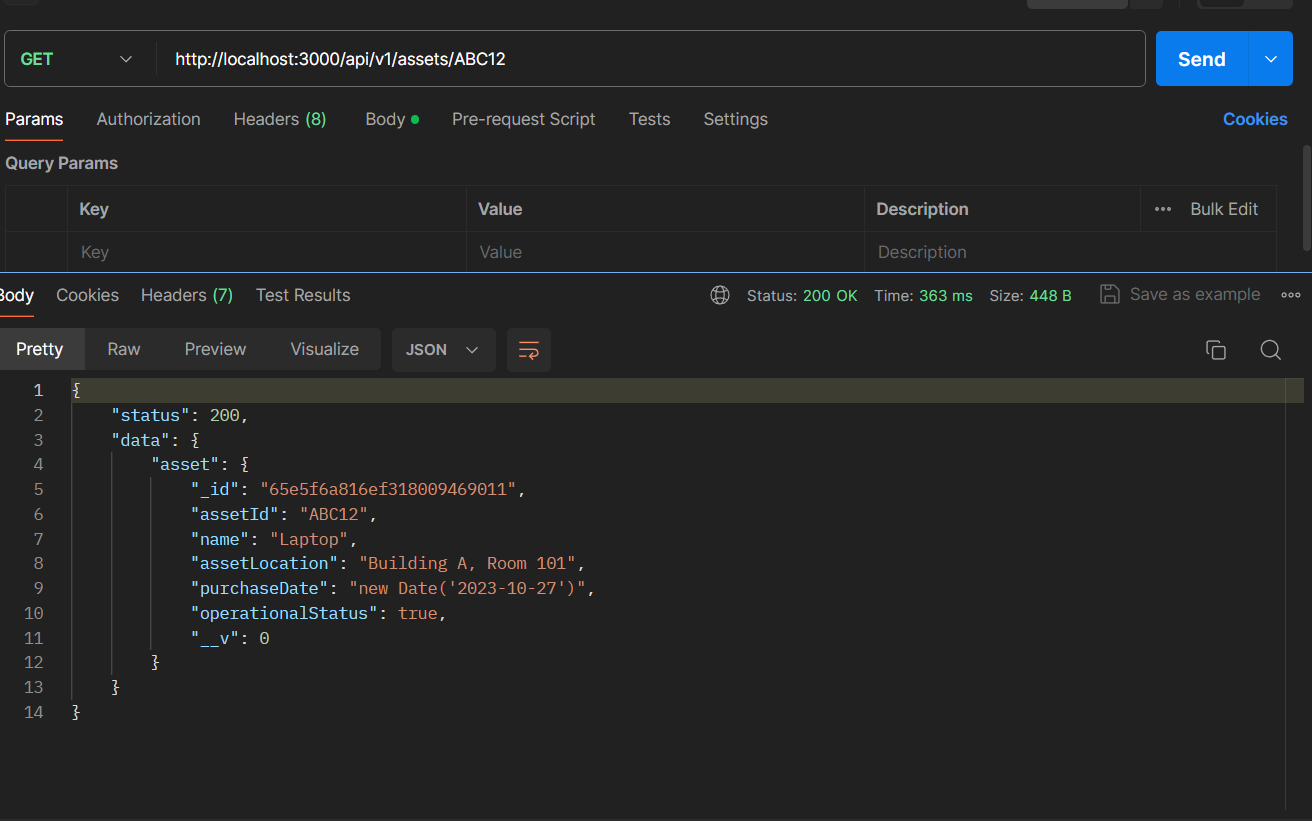
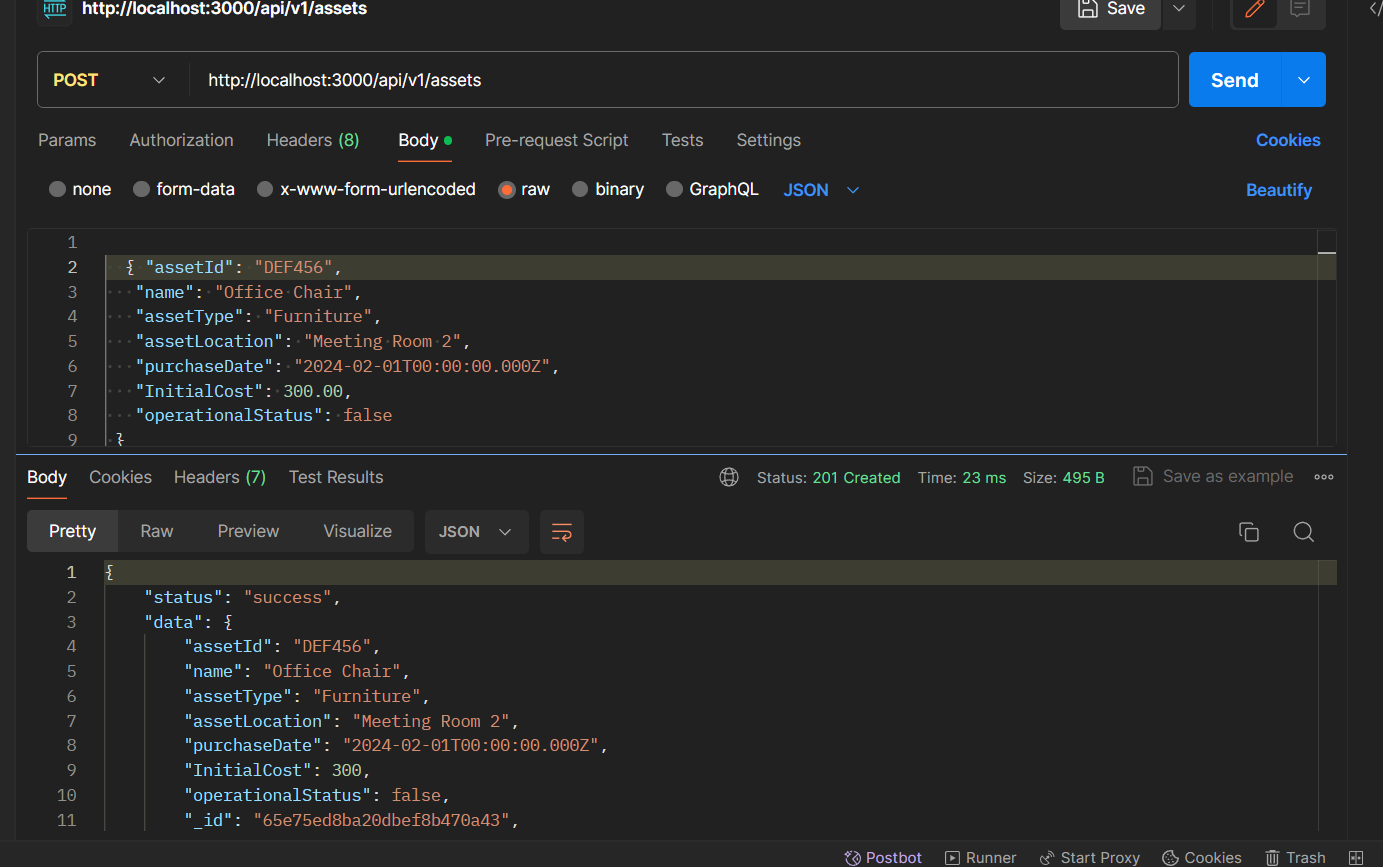
The project was not a straight forward for me. Implementing the APIs was not a big issue. The database design was something which struck in my mind that the two separate collections can be created and using the primary key of asset that is ASSET\_ID and using it as foreign key for PERFORMANCE metric. This allows us to directly map the object t its performance metric.   
There was API functionality for every operation but the display of data based on lower runtime was not been implemented. I was stuck on how to implement it and tried my best to complete it within my designated time . The authentication part was implemented with the help of MongoDB and the users were stored in database and a checking was done for the secured route that user was present or not in our database.

Let’s see the GET request when an asset is provided for the document.



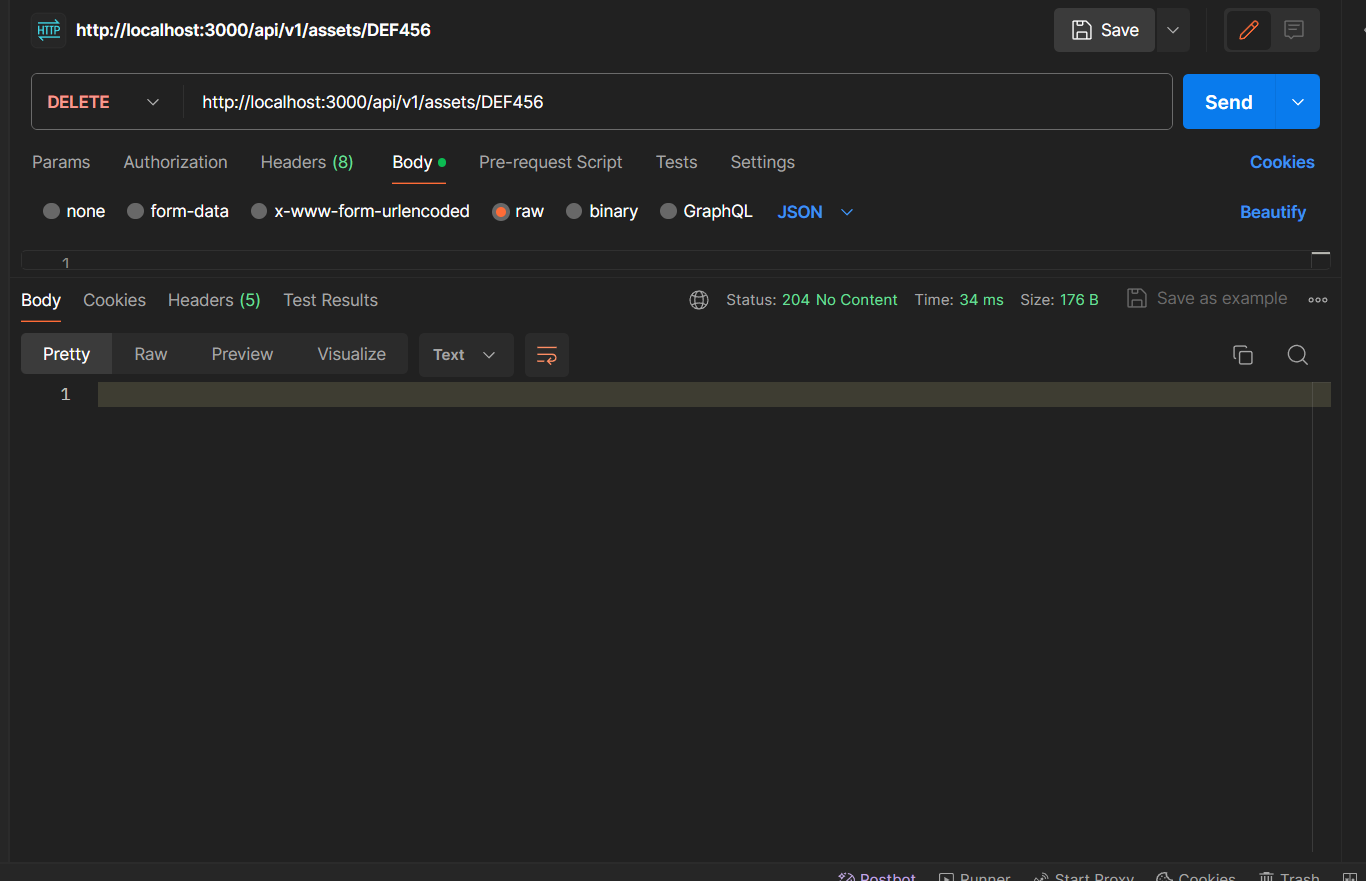
POST A DATA TO ASSET COLLECTION.

Now let’s see another post request.  
The data is provided in the body of request and from there it is extracted and posted to MongoDB Assets collection.



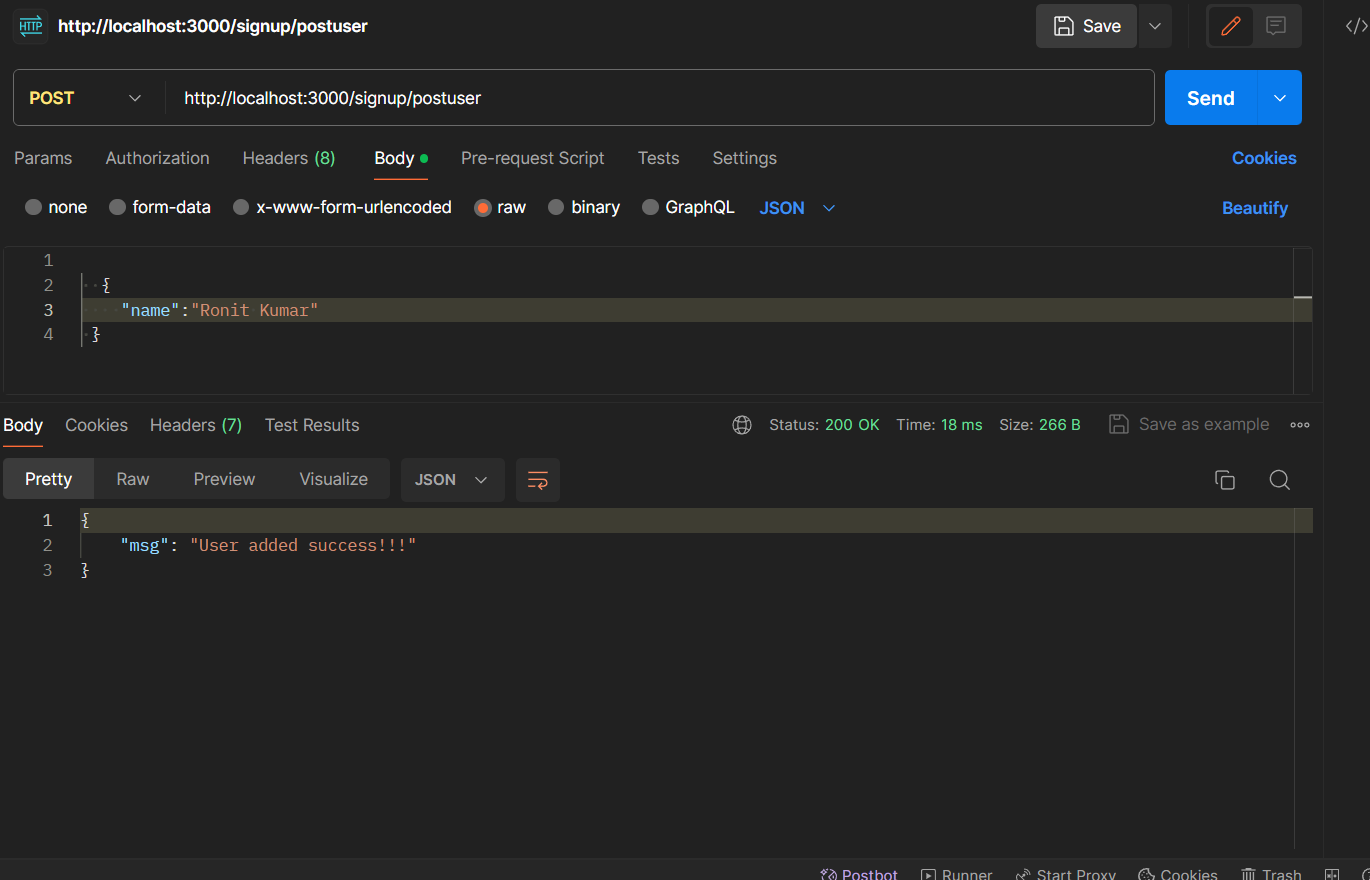
DELETE A DATA FROM ASSET COLLECTION.

We use functionality of delete and using the asset id we delete a particular data from the MongoDB database. The user has to provide the assetID in the URL and that particular asset would be deleted.

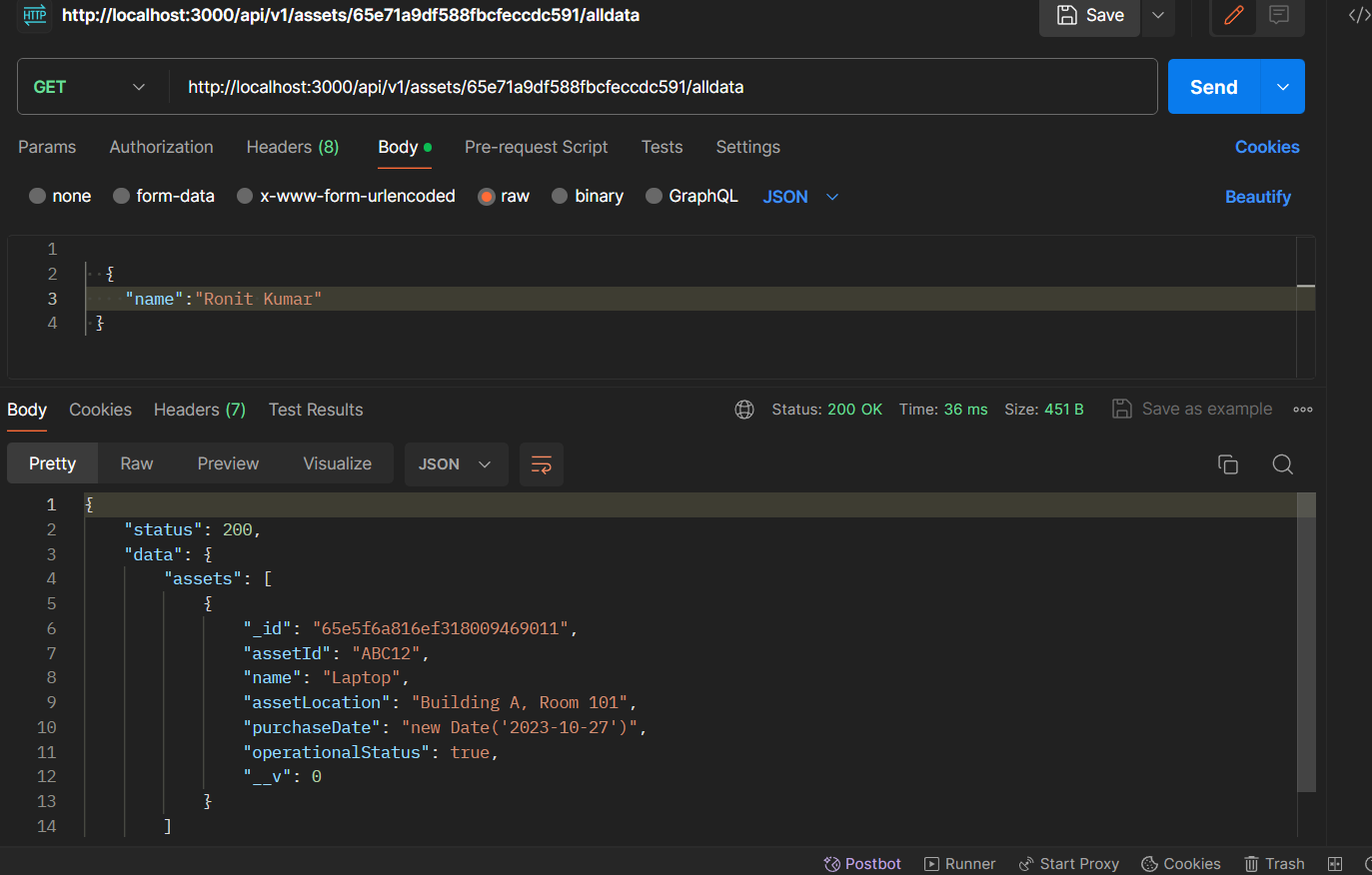


USER ROUTE POSTING A USER DETAILS

Now let’s see for the user route. First the user details must be present in the database. So, the user details have to be posted to the database from where it could be checked that the user was present or not.



THE AUTHETICATION PART   
  
this was a bit complex section as I had to know the user through the id given to it from the MongoDB and if it was a valid USER than only show the data to the user.

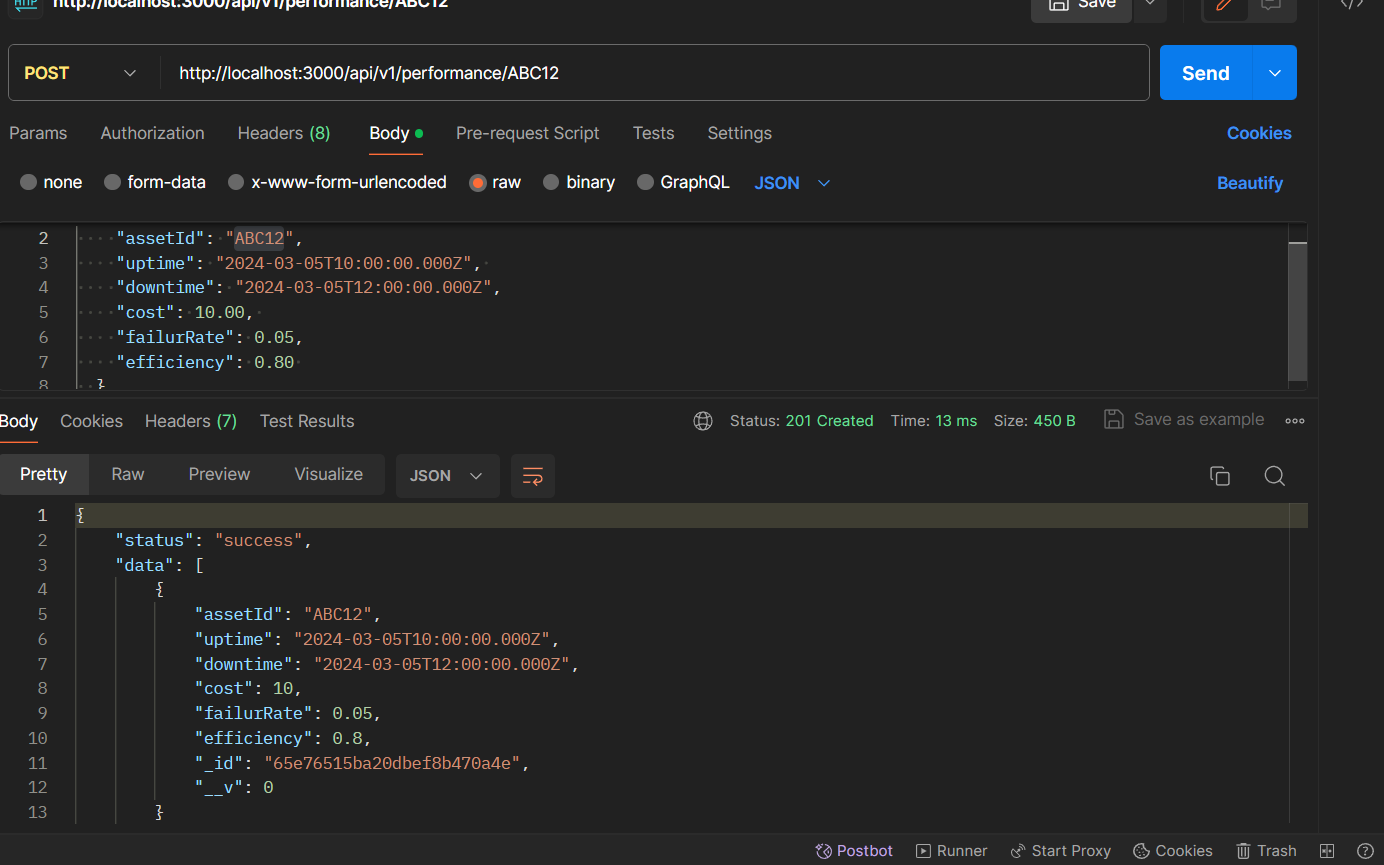


As you can see in the URL the userID was provided and if the userID will be correct than only the information will be displayed to the user.

PERFORMANCE METRIC

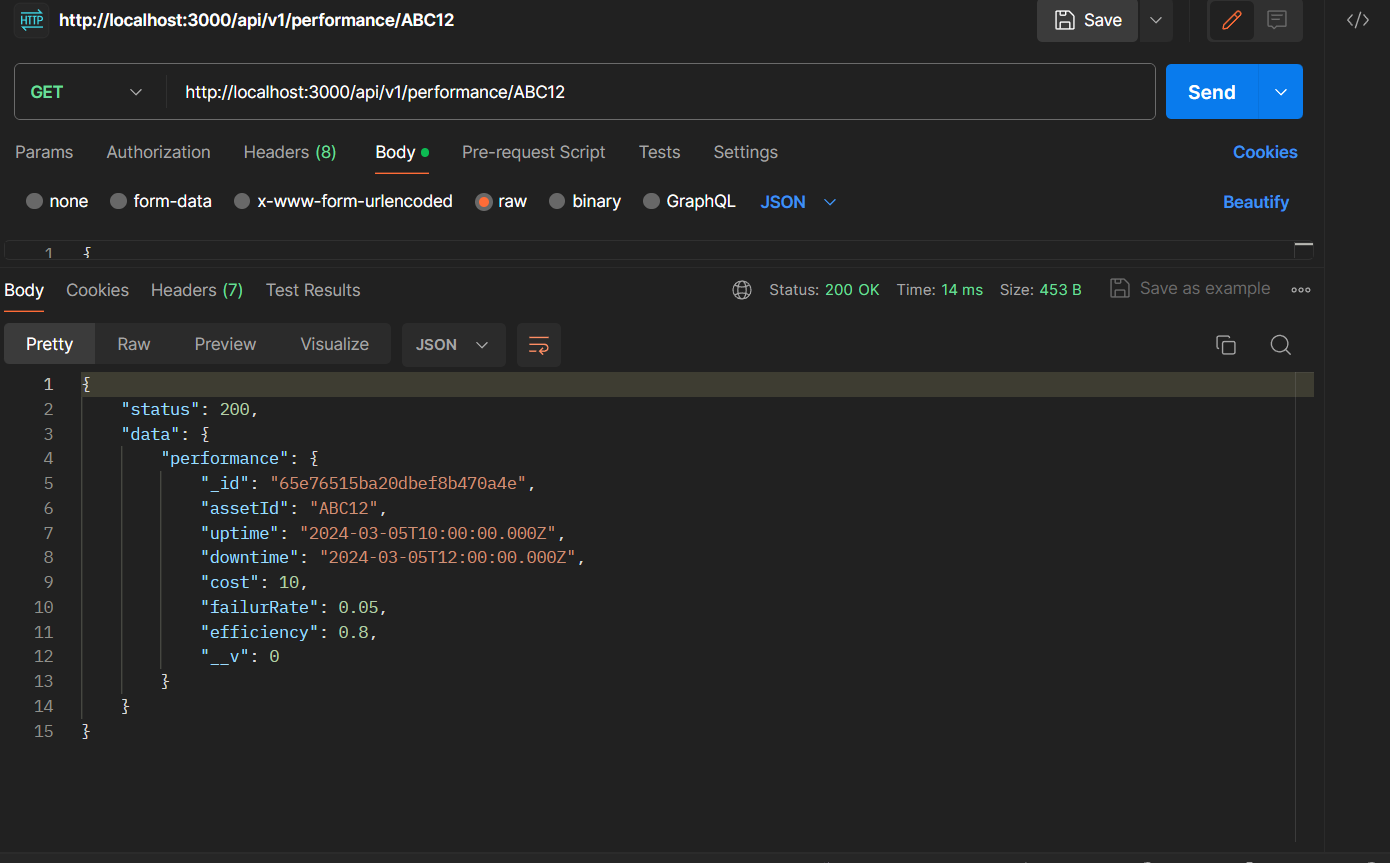
The post request for performance metric is used here. The assetID was provided in the URL and if the asset with given id is present in the asset collection, then only the performance analytics will be posted.

Even duplicate data will be scrapped off and will not be added .

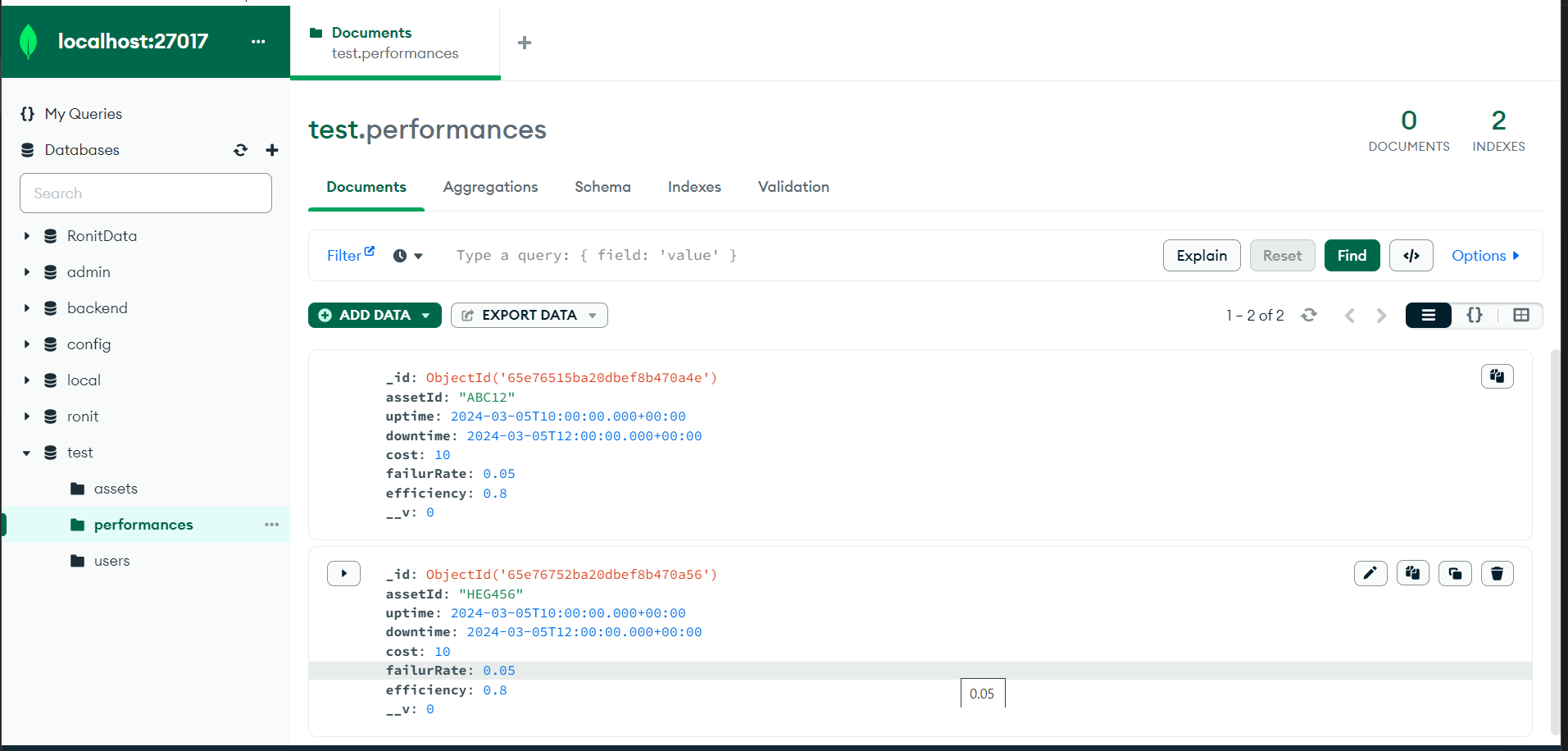


GET REQUEST FOR PERFORMANCE OF ASSET

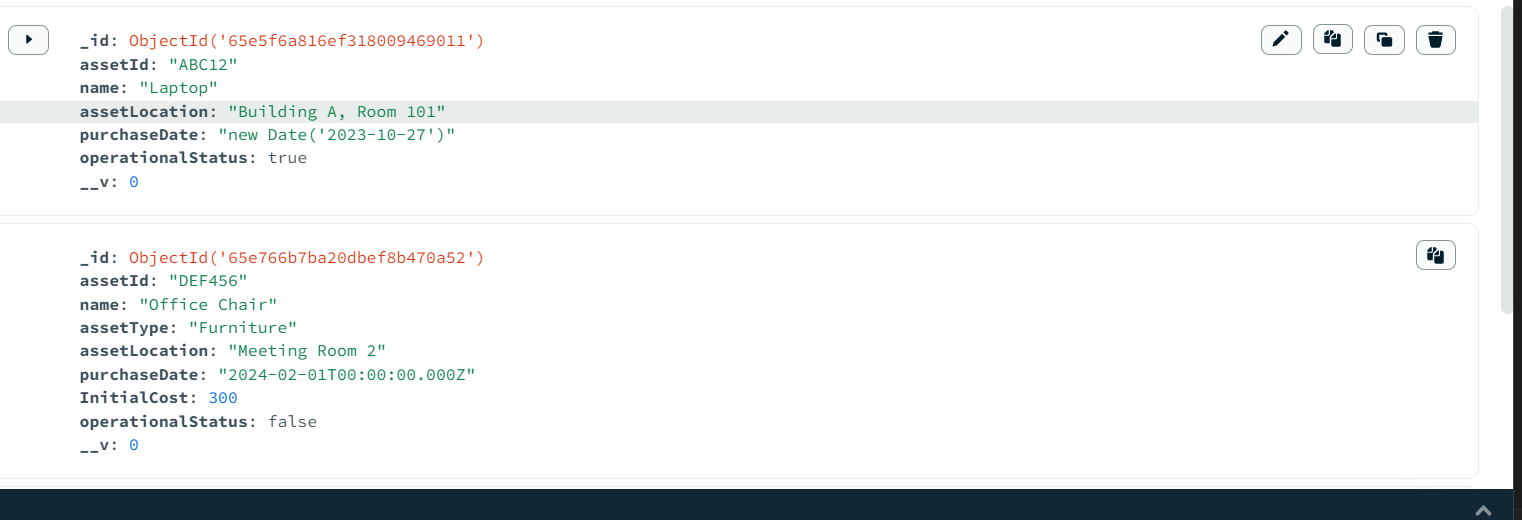
Now if we want to get the get the performance analytics of the data then the GET request with URL containing the assetID for which you have to get the data. So, if the assetID is not correct an error response would be shown. If the assetID is correct then we would get a proper response. Let’s see an example about it.

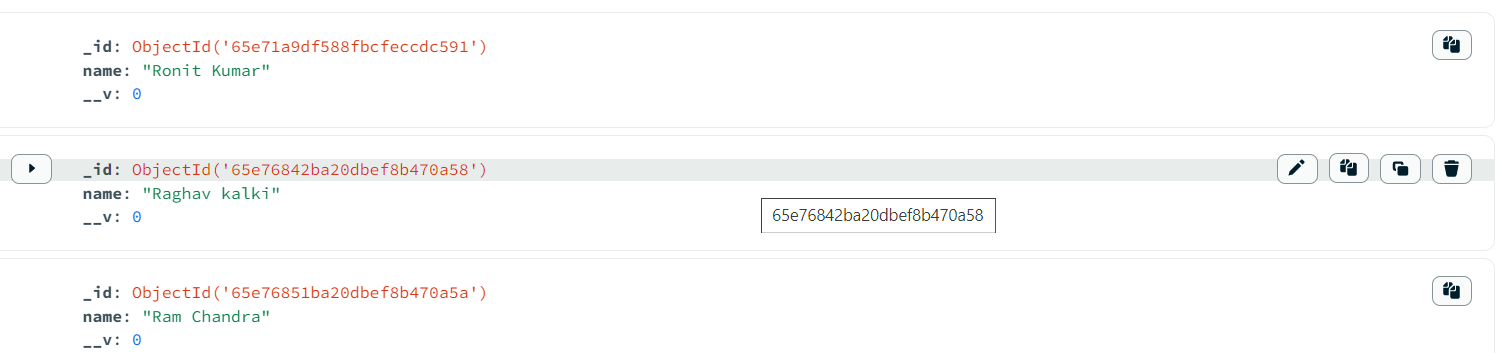


PERFORMANCE MODEL WITH ALL THE DATABASE IMAGE SHOWN HERE.



ASSET MODEL





These are all the model and the way they are being stored here in our MongoDB database .  
  
They are a few other APIs as well which are running all fine and can be implemented in similar way.  
  
This project taught me various things like how to have more than one collection and how to interlink them so that the data from can used to get the element from other collection. How to use authentication so that some particular route is protected and can be encrypted. How to implement various techniques within the APIs was also something which made learn more.  
  
All in all it was absolutely wonderful experience to make this project and learn loads and loads of thing and would definitely like these challenging things to work upon in the future.  
  
Thank you   
Ronit Kumar Baranwal   
CSE – JADAVPUR UNIVERSITY