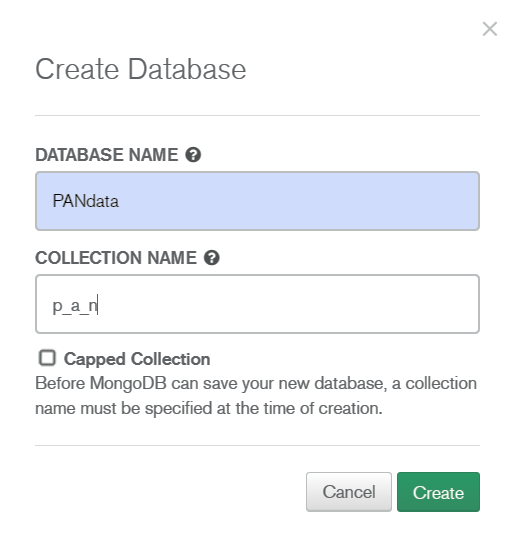
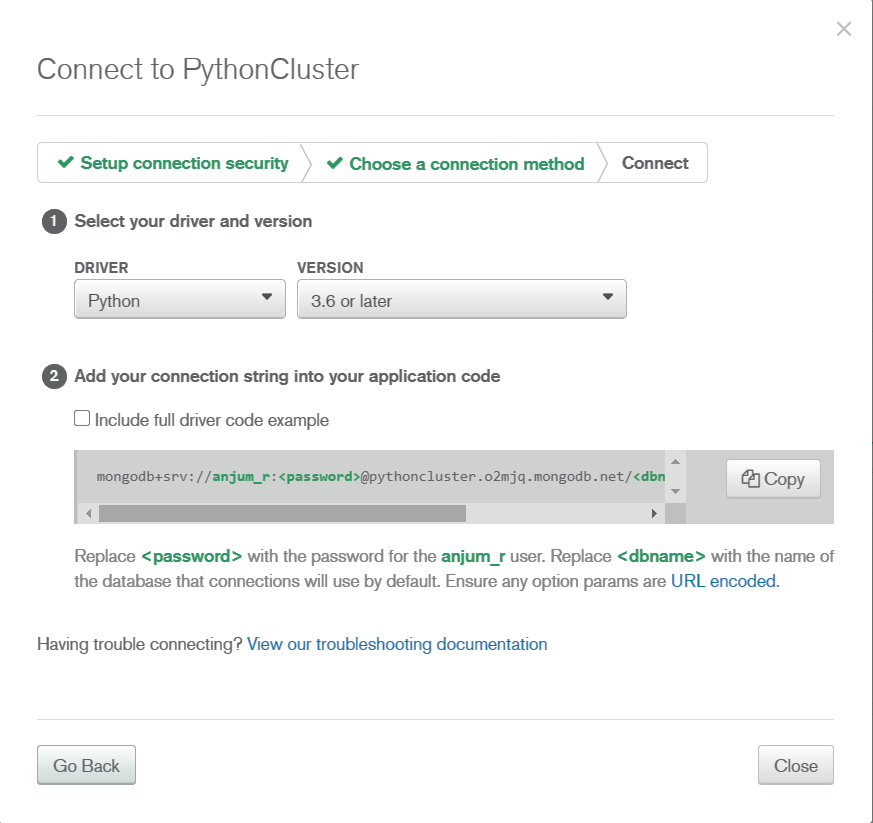
**Task Summary**

1. The first step is to create a MongoDB database, where the Python Cluster is created consisting of Database and its collection, corresponding to a particular user.

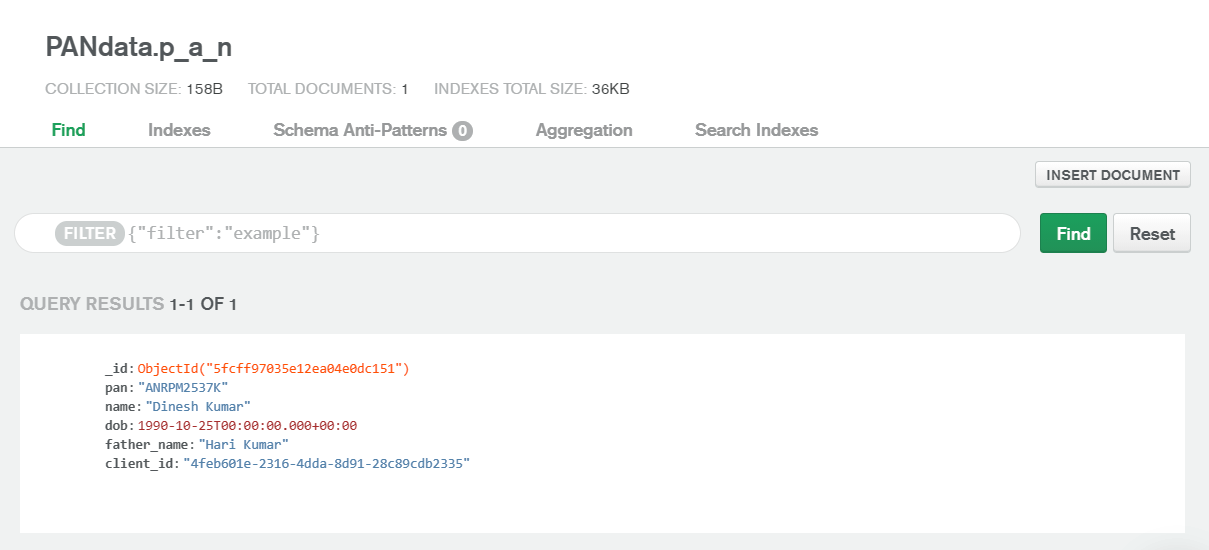


As seen in the above figure, the Database name and Collection name are assigned and the database is created with the name “PANdata”.

1. After creating the database, the second step is connecting to the database server. This is done by using “db.connect()” command. For establishing connection, it is important to connect to the generated Python Cluster. For this, a connection string is generated which is to be fed to the application code as DB-URI. This is shown in the figure below:

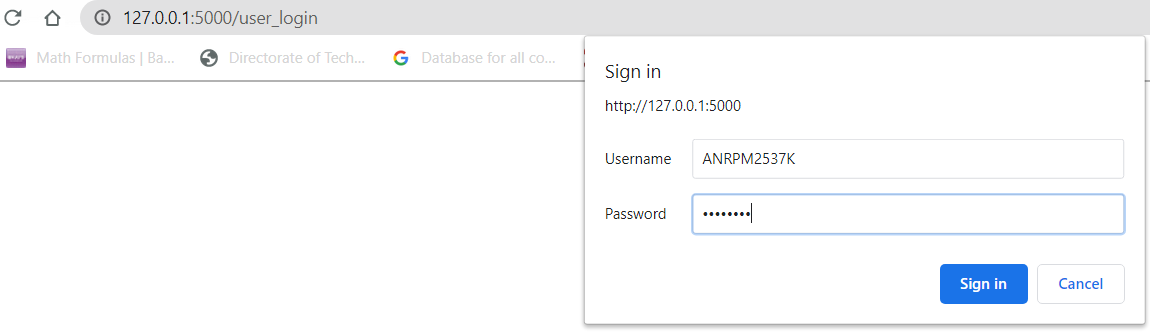


1. After establishing connection with the MongoDB server, the details can be fed to the database collection, named as “p\_a\_n”. Here, the PAN data associated with the given PAN number is stored in the database as shown:

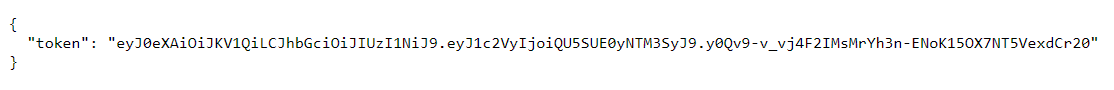


1. After storing the details in the database, the next step is creation of API endpoints. According to the task, 3 endpoints are created:
2. Generation of JASON Web Token (JWT) for authorization:

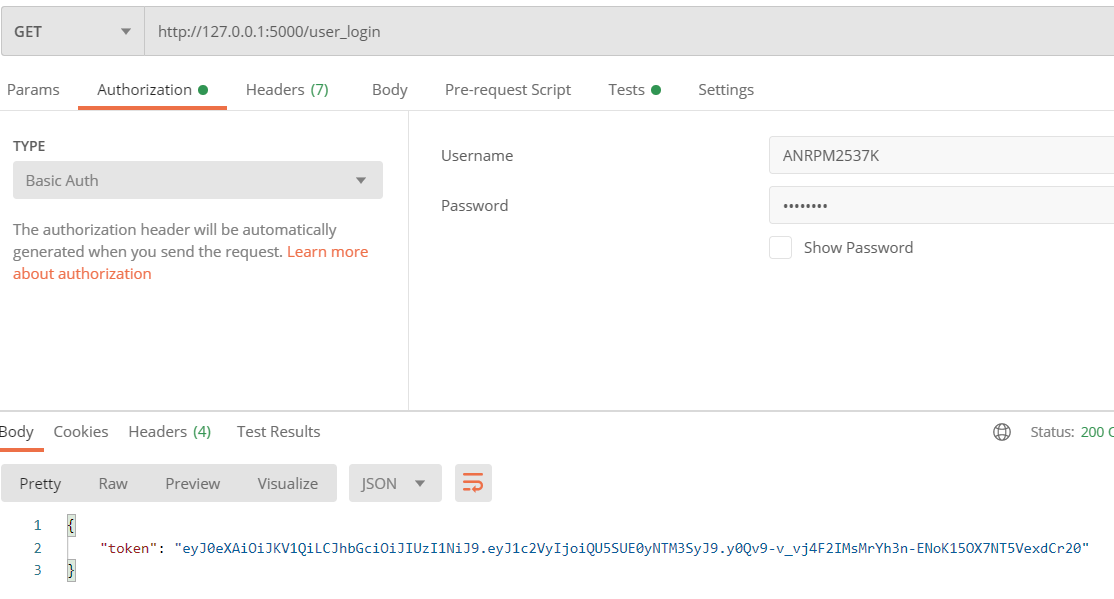
JWTs are generated using a secret key, which is used during both encryption and decryption of the token. The token is generated only if login is successful. For successful login, correct PAN number should be entered in the “Username” section along with the correct password as shown:



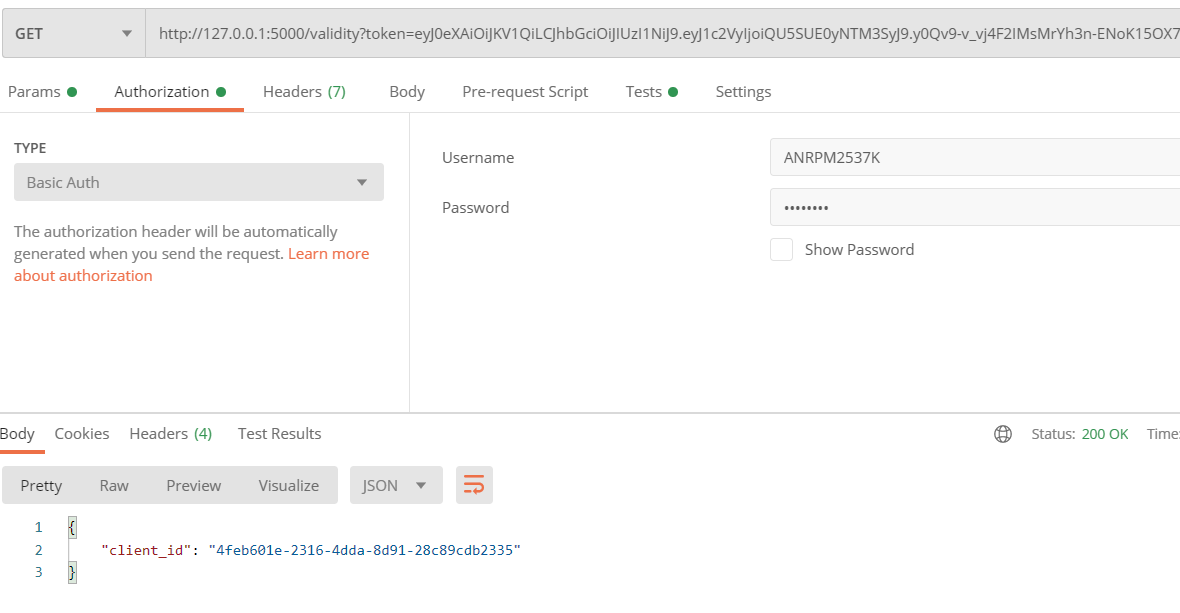
After successful Sign in, encoded token is generated as shown:



This can also be implemented on the “Postman” API testing tool:



1. Now, in the second endpoint, if the above generated valid token is entered, the client\_id corresponding to the input PAN number is generated as shown below:



1. Finally, the third endpoint gives all details (in json format) corresponding to the client\_id and PAN number after entering the correct token as shown below:

