**MFLIX**

Movies streaming app

**Jyothish Kumar CHANDRASENAN GEETHAKUMARI**

**Msc Data Science and Analytics**

**Fall 2019- 2021**

**12-June-2020**

**Overview**

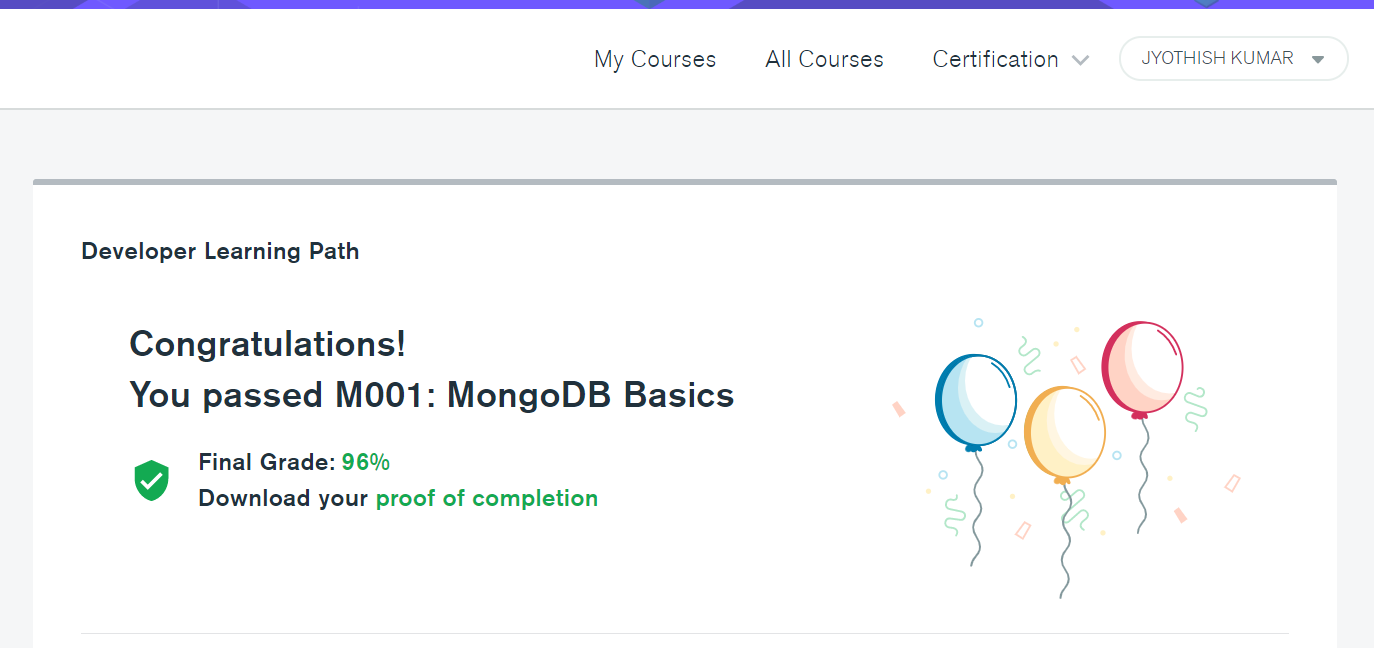
MFlix is a movie streaming application which uses MongoDB as the database in the back-end.

Mflix is composed of two main components:

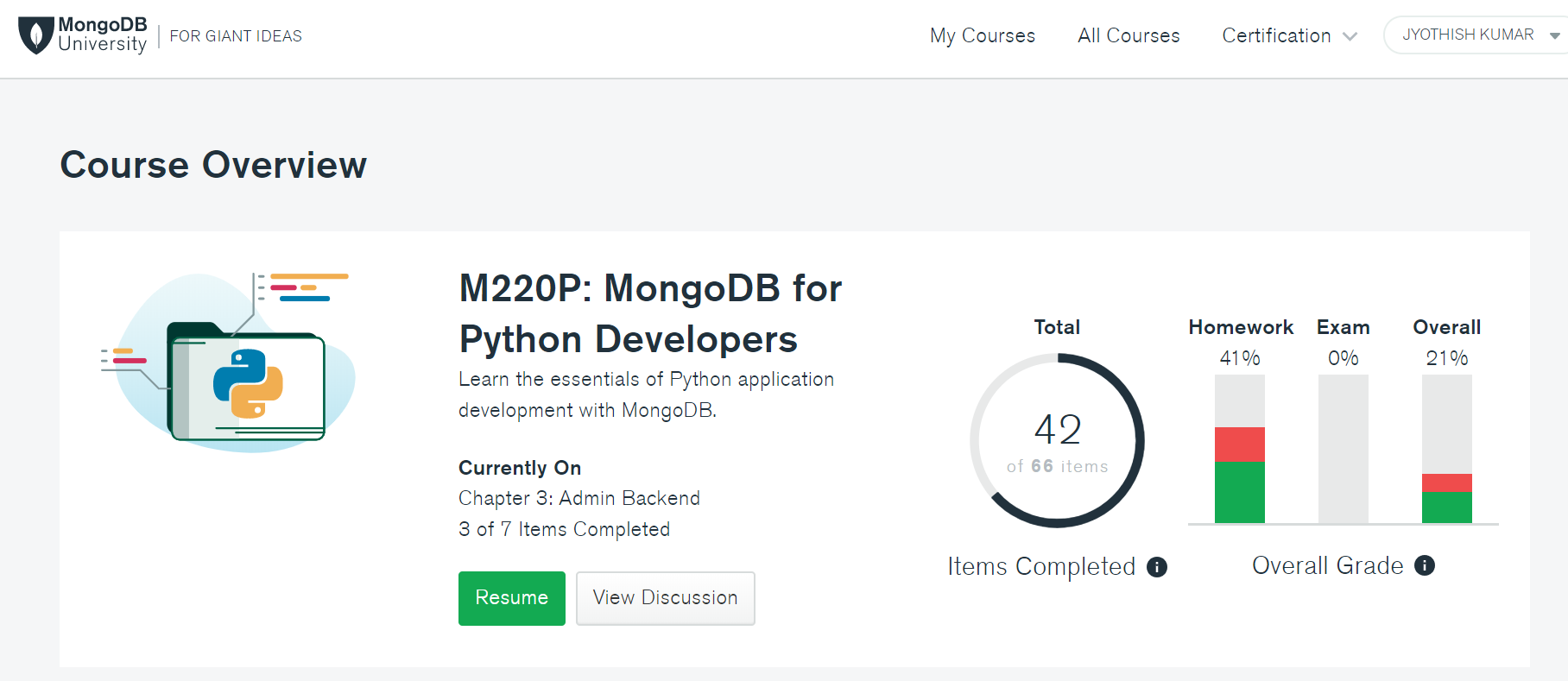
* *Frontend*: All the UI functionality is already implemented which includes the built-in React application.
* *Backend*: The project that provides the necessary service to the application. The code flow is already implemented except some functions.

I have taken the below two courses in <https://university.mongodb.com/> for learning MongoDB and implemented concepts in this project followed by the university.

* M001: MongoDB Basics



* MongoDB for Python Developers (In Progress)



Code is uploaded to Github : <https://github.com/jkcg-learning/MFlix>

**Tools Used**

* MongoDB Compass – GUI interface for interacting with MongoDB
* Mongo shell – Shell mode for interacting with MongoDB
* MongoDB Atlas - Cloud Database
* Necessary Python Libraries (specified in the requirement.txt file)

**Creation of Cluster in the MongoDB Atlas**

* Create an Organization

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

* Create a Project & a Cluster

A picture containing screenshot

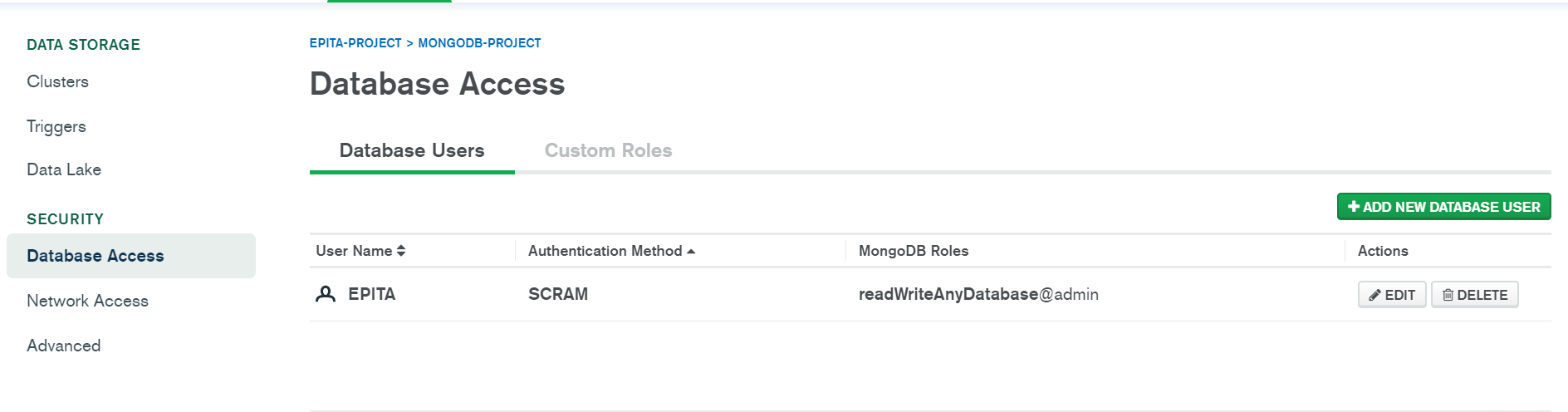
Description automatically generated

By Default Mflix database is loaded into the cluster

A screenshot of a cell phone

Description automatically generated

* Provide Database Access and Network Access for the Cluster



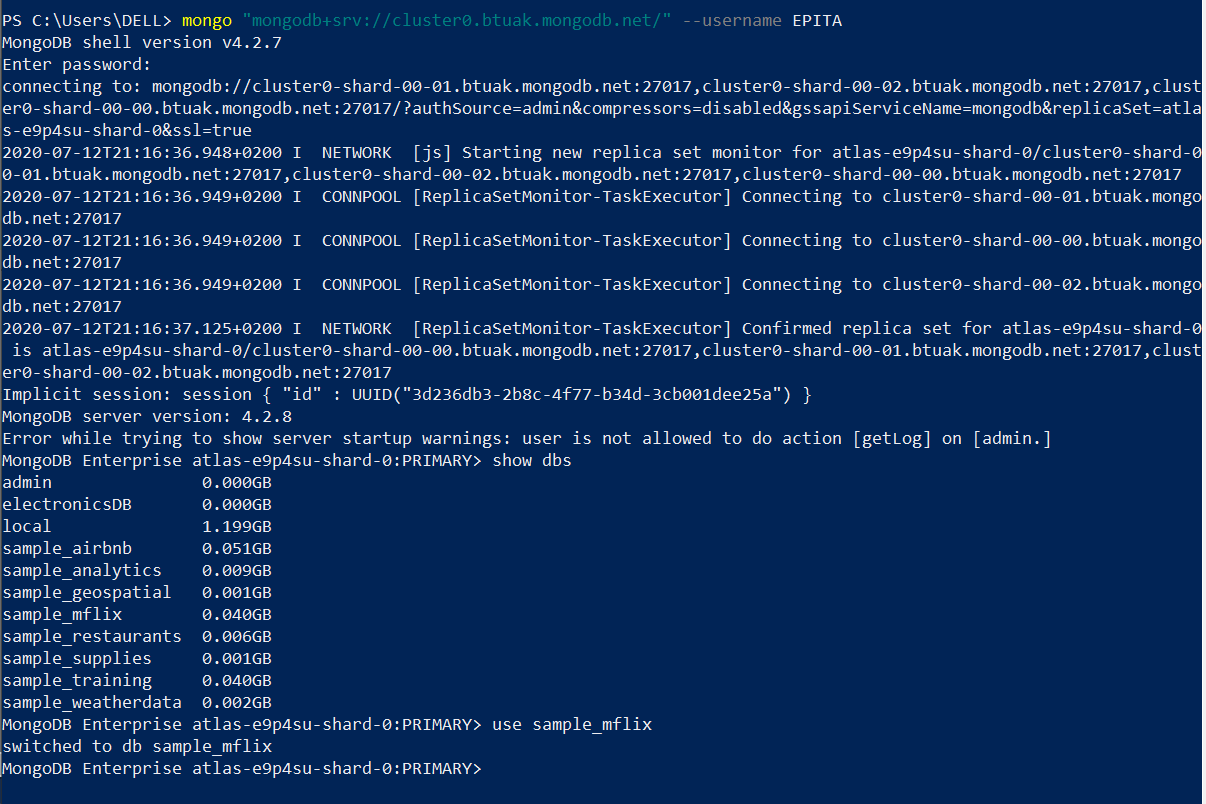
**Connect cluster using Mongo shell and MongoDB Compass**

**Mongo Shell**

**mongo "mongodb+srv://cluster0.btuak.mongodb.net/" --username EPITA**

A screenshot of a cell phone

Description automatically generated

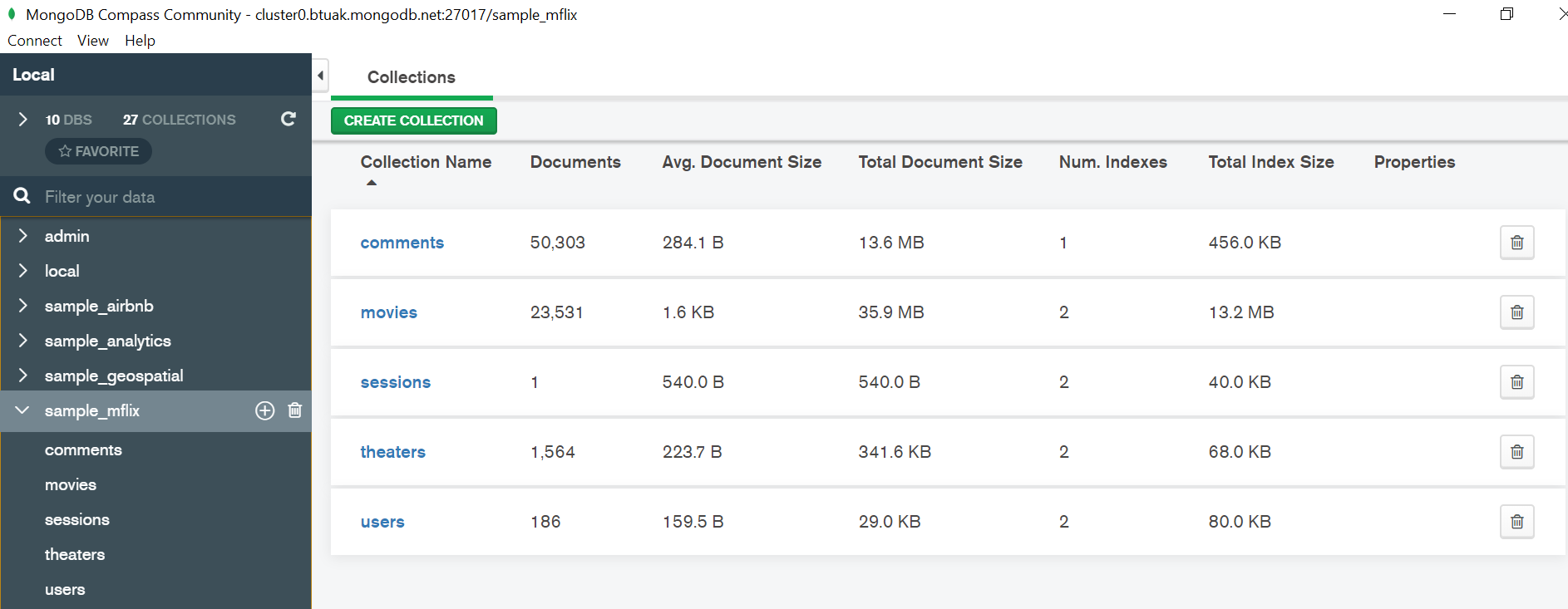


**MongoDB Compass**

**mongodb+srv://EPITA:<password>@cluster0.btuak.mongodb.net/**

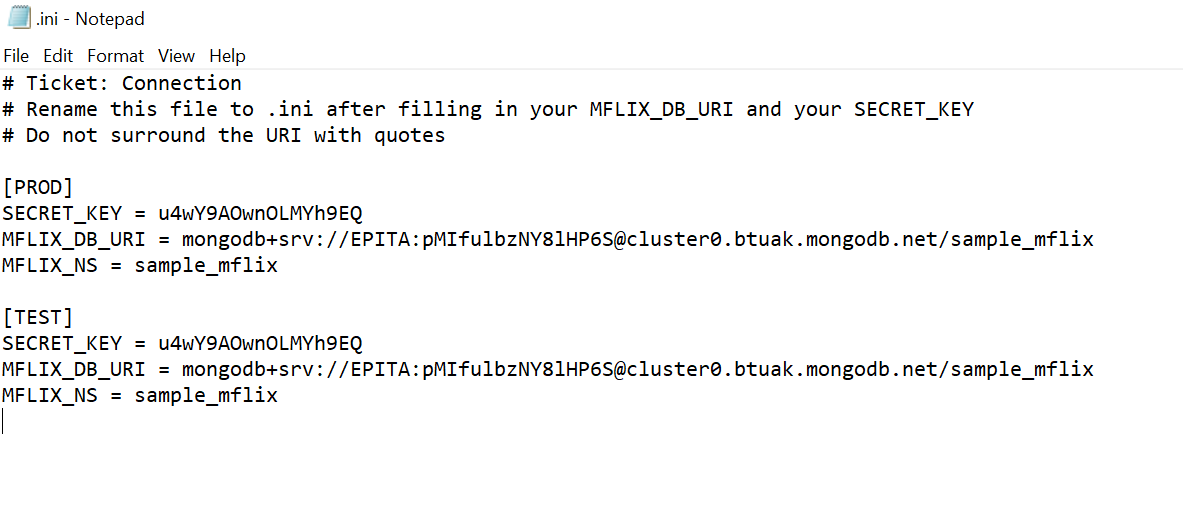
**A screenshot of a cell phone

Description automatically generated**



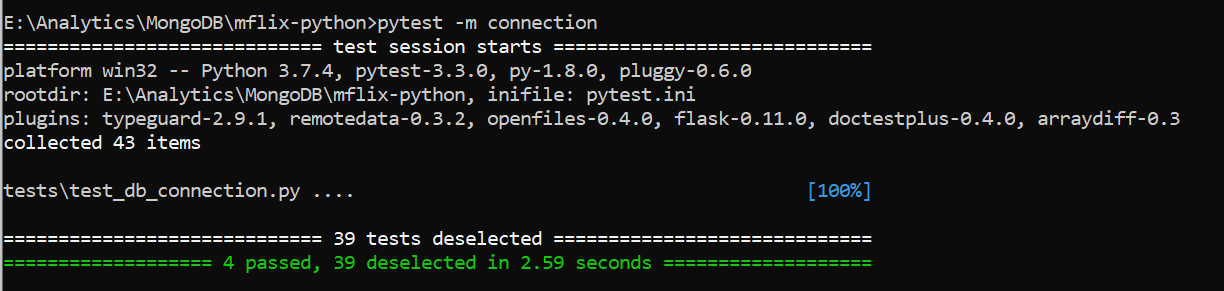
**Launch the MFlix Application**

* **Download the project from the university**
* **Make the necessary changes in the .ini file which will connect the application to the database in our cluster.**



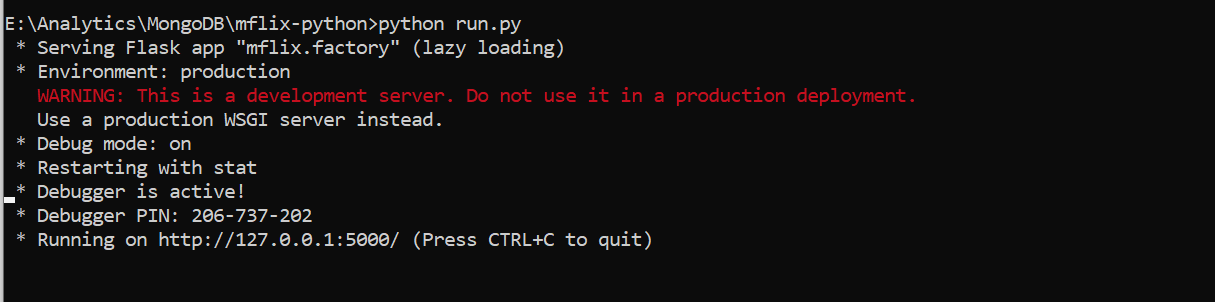
* **Test the connection**

pytest -m connection



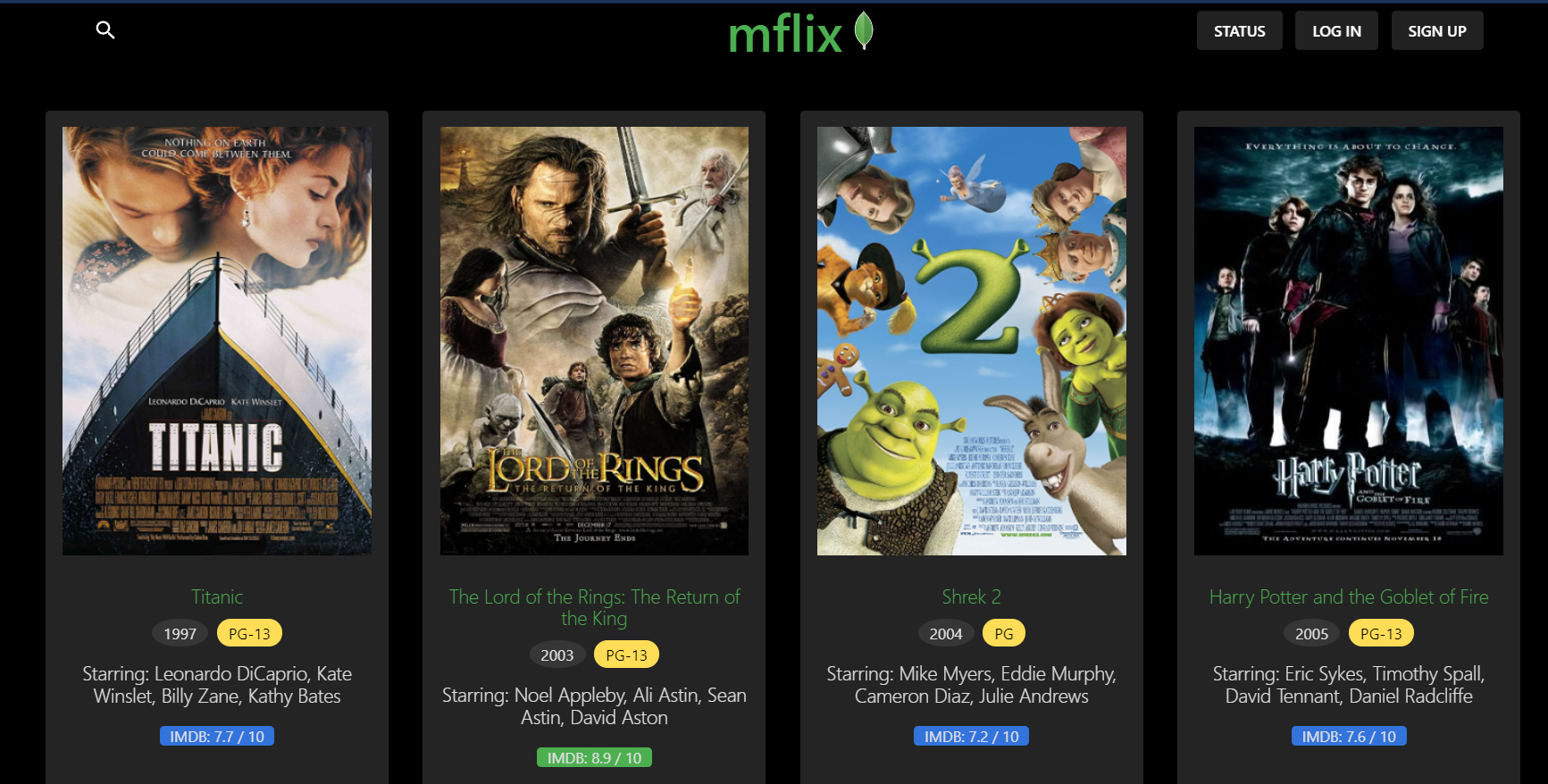
* **Run the application**

python run.py



Application is launched and can be accessed using

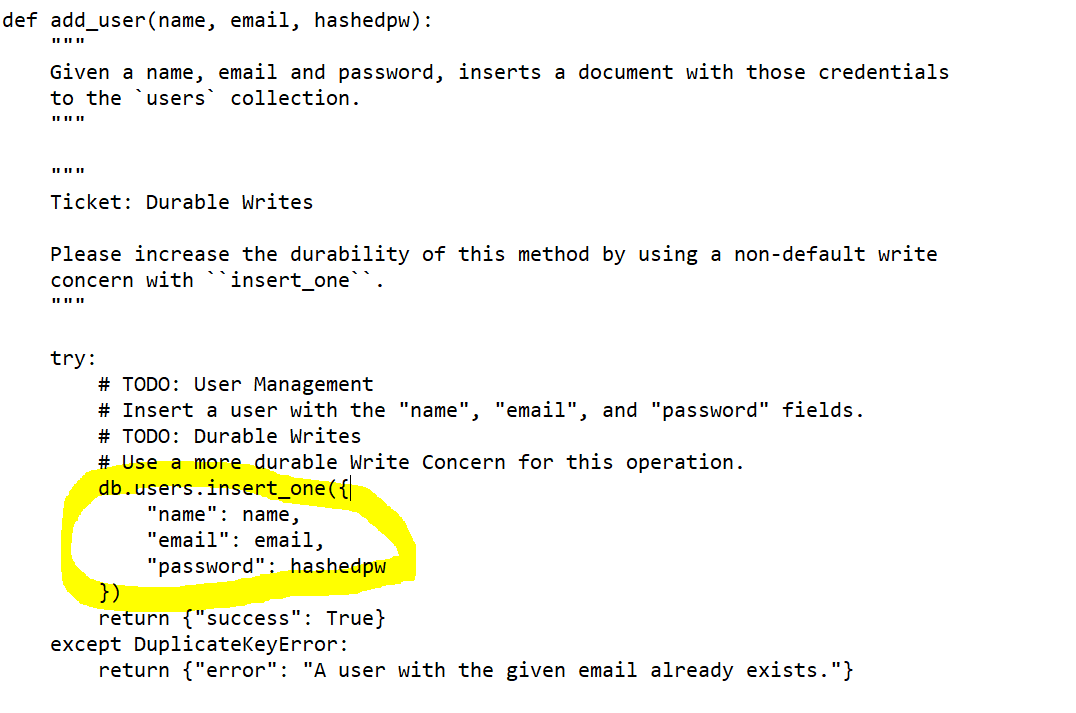
<http://127.0.0.1:5000/>



**CRUD Operations**

* **CREATE**

A new user can be created using the front-end and back-end function is written to insert it into MongoDB



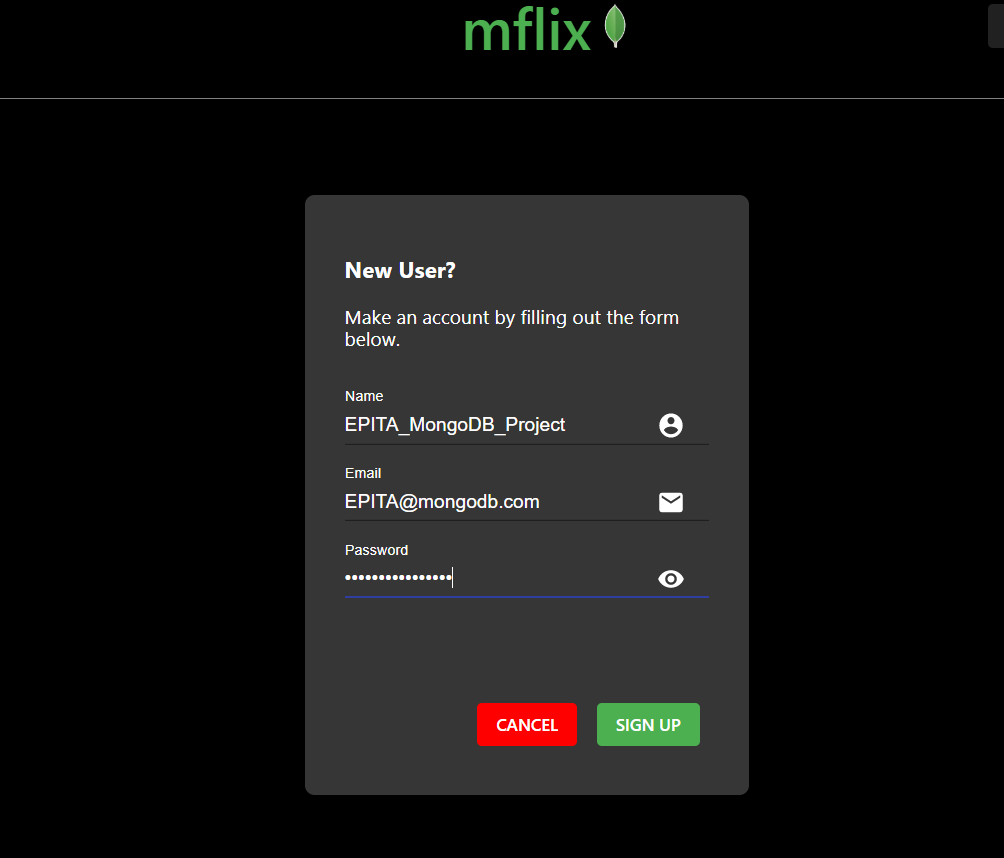
**Using collections.insert\_one() from MongoDB**

db.users.insert\_one({

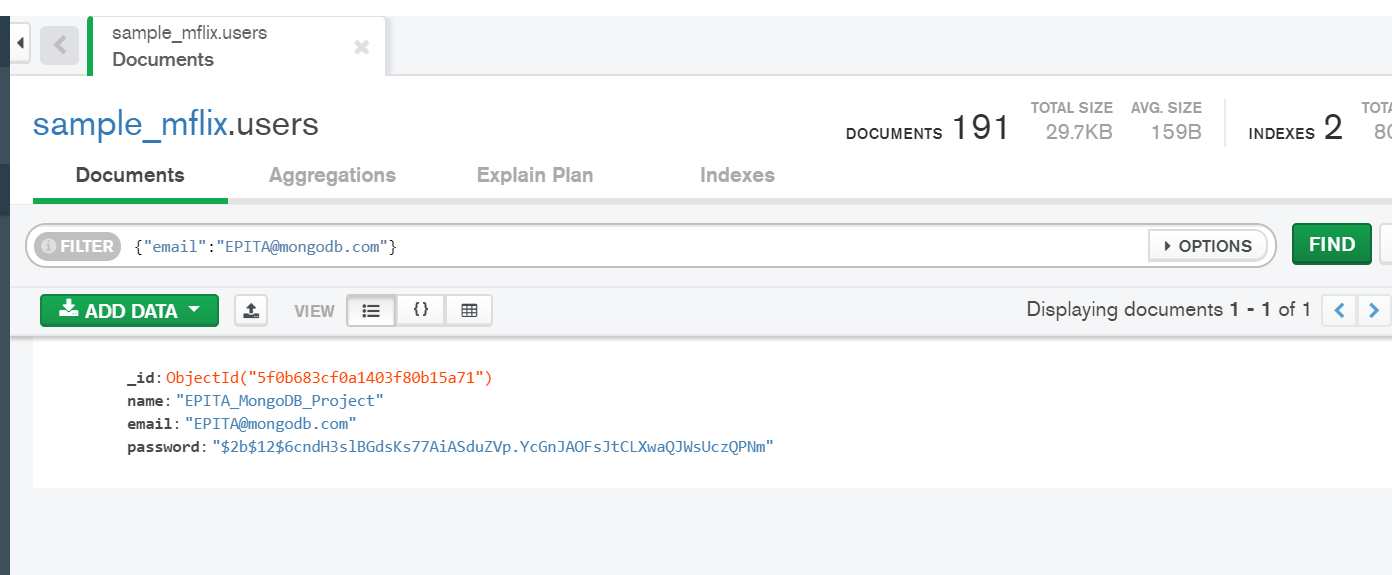
"name": name,

"email": email,

"password": hashedpw})

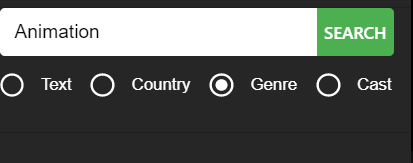


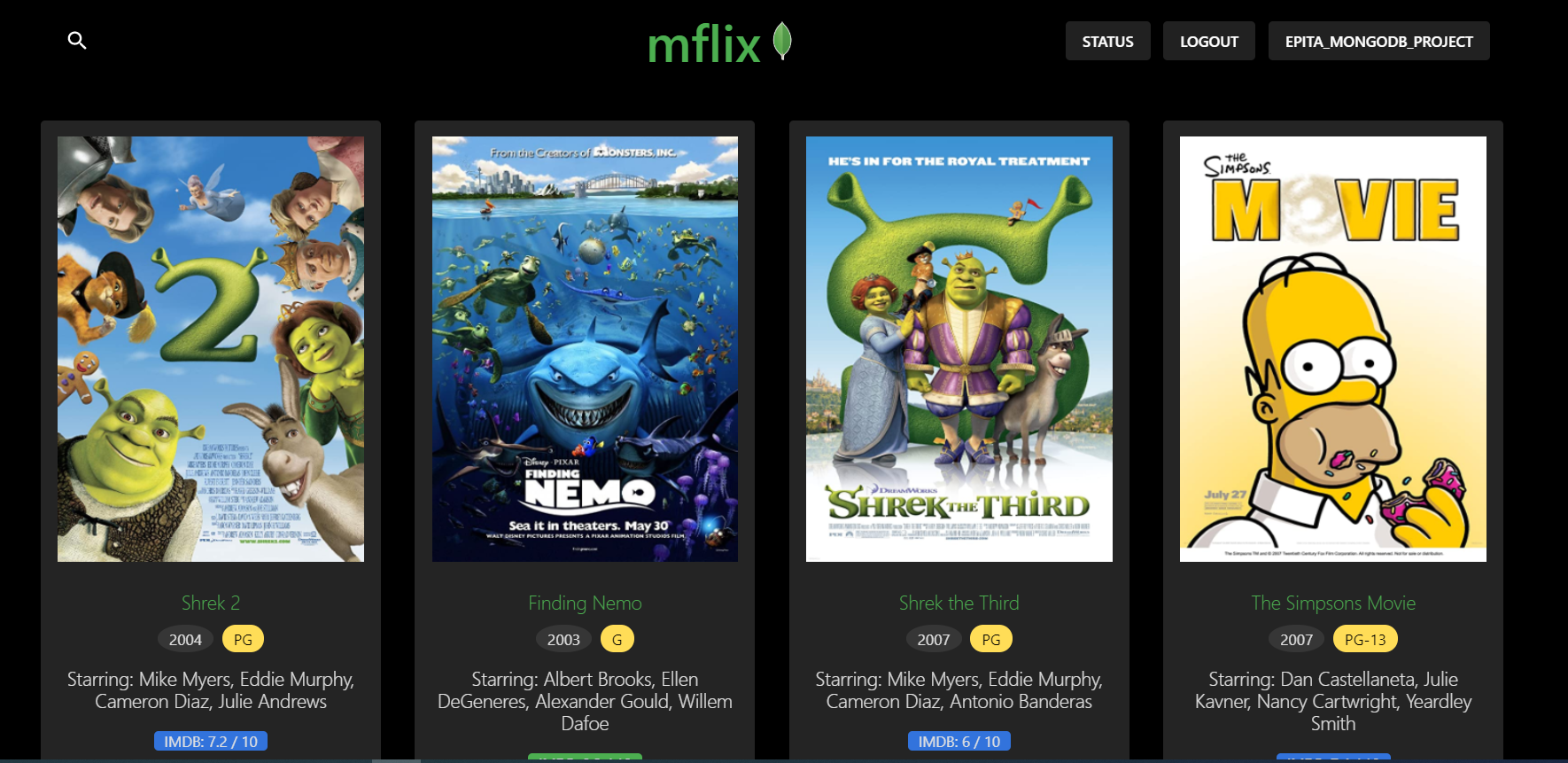
The details are inserted in the users collection of sample\_mflix database



* **Read**

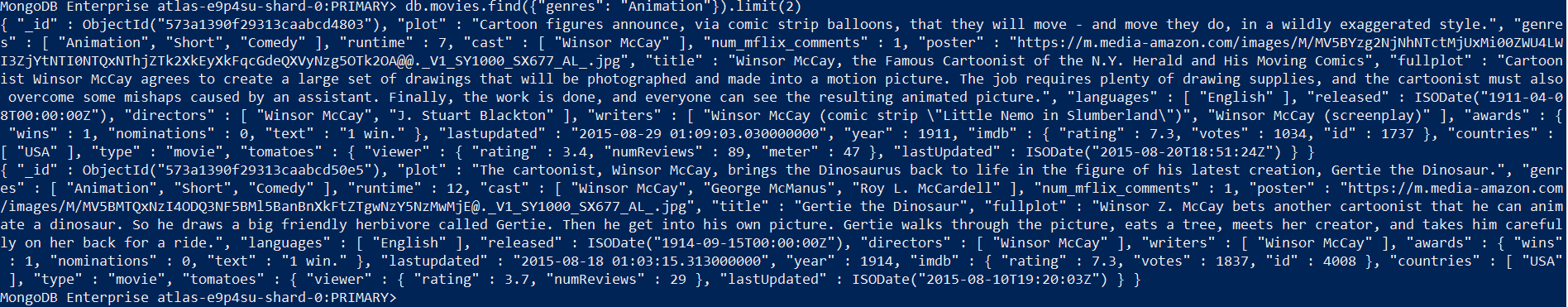
Search for movies based on Genres

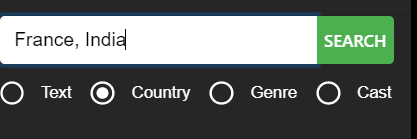




**Using collections.find() from MongoDB**

db.movies.find({"genres": "Animation"}).limit(2)

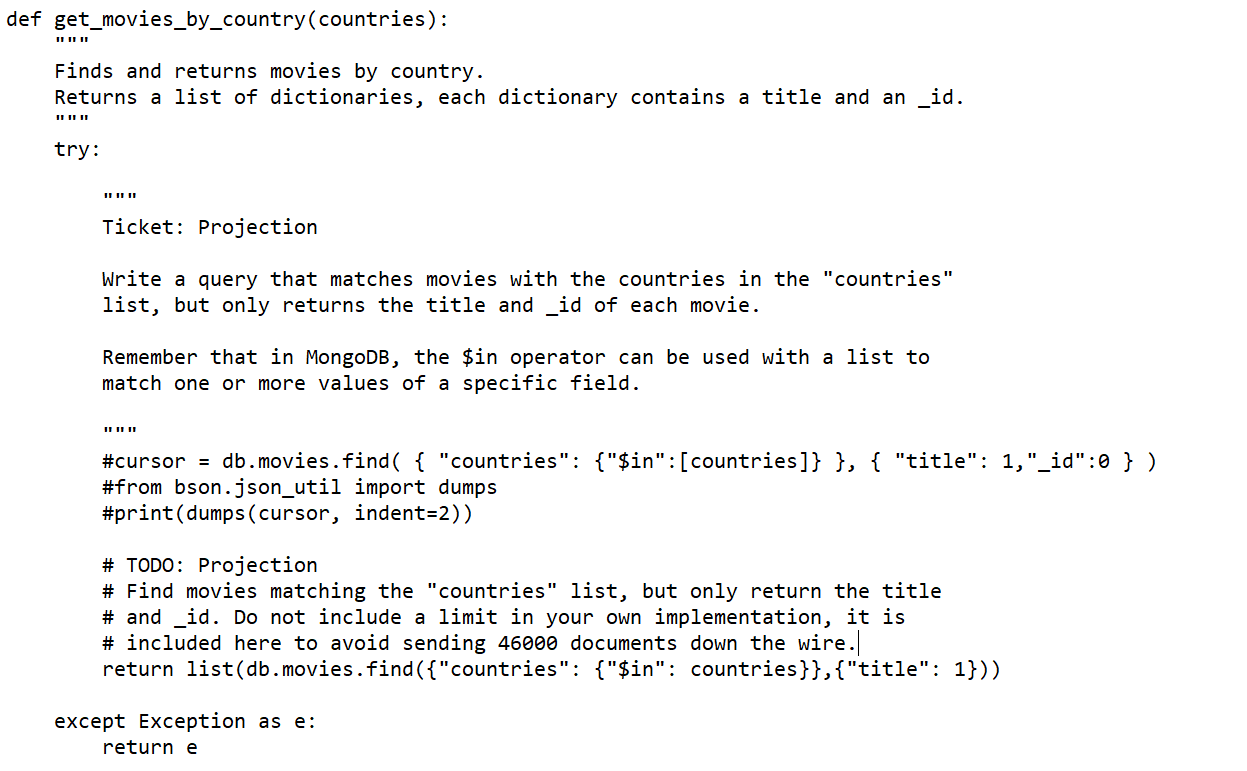






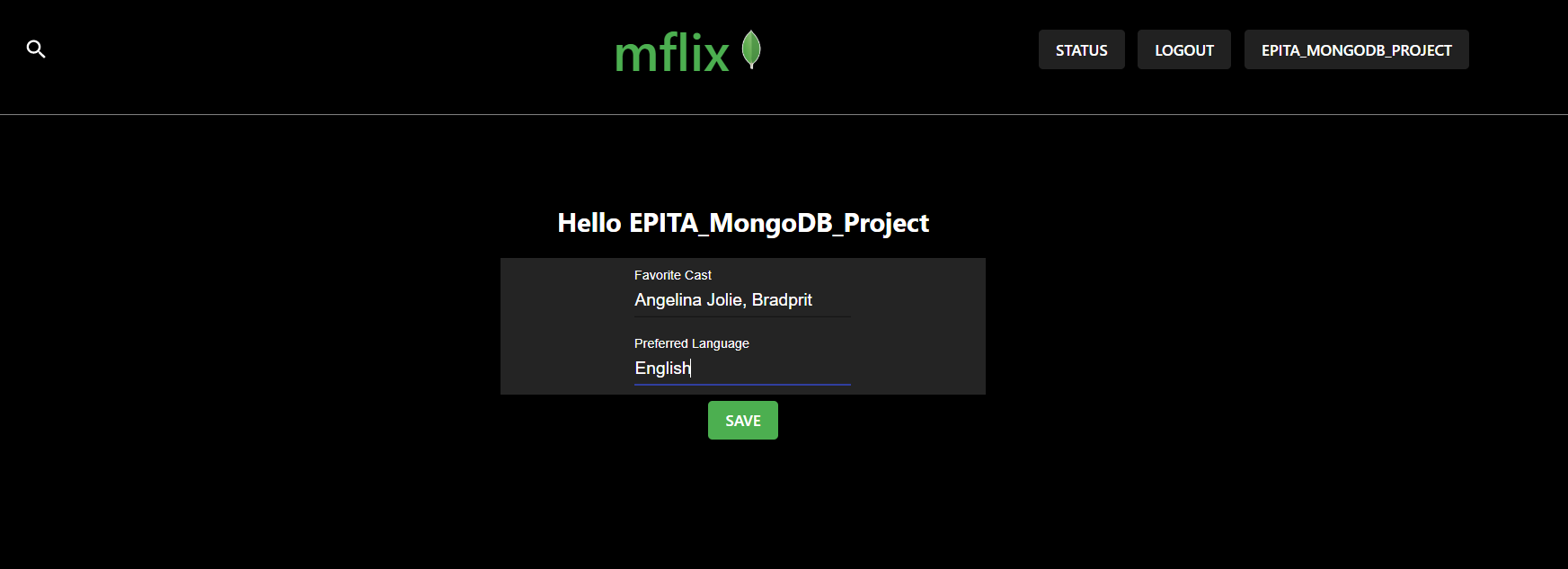
db.movies.find( { "countries": {"$in":[countries]} }, { "title": 1,"\_id":0 } )

Only Titles are displayed – “titles”:1



* **UPDATE**

Adding the preferences of the registered users. This update the users collections with new informations.



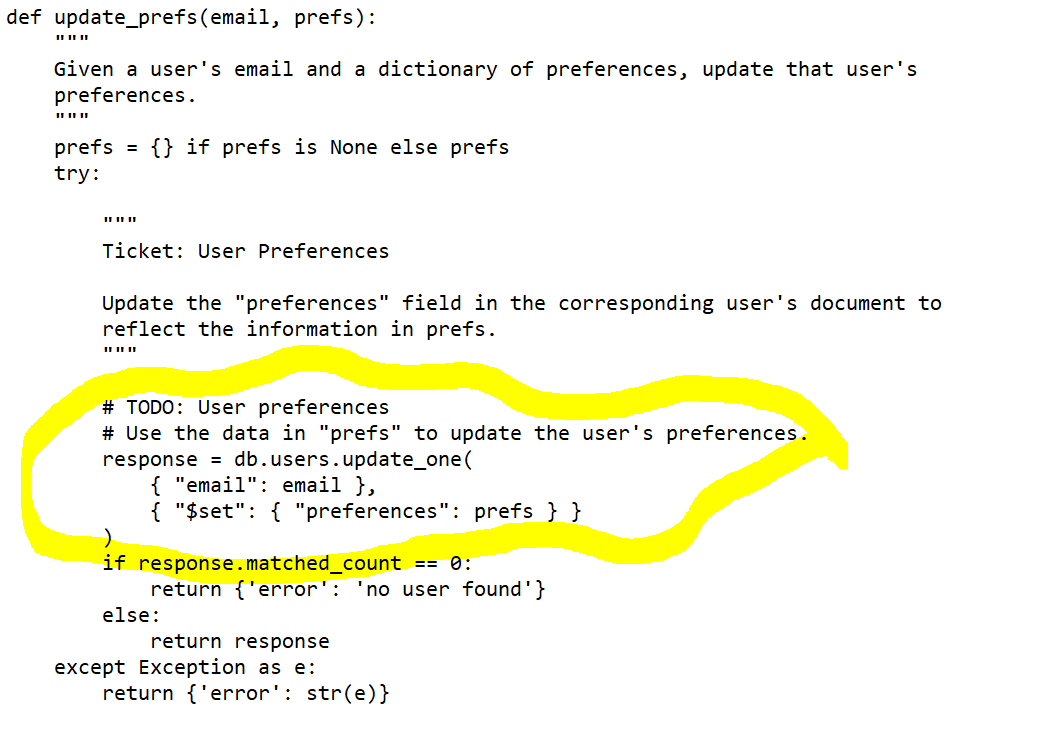
**Using collections.update\_one() from MongoDB**

db.users.update\_one(

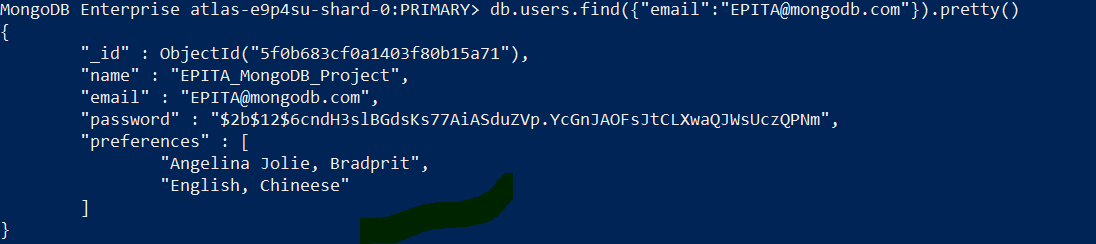
{ "email": email },

{ "$set": { "preferences": prefs } }

)



The details are updated in the users collection of sample\_mflix database

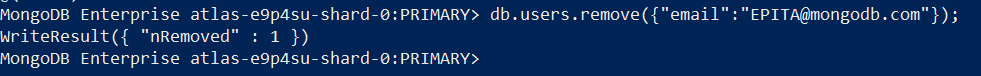


* **DELETE**

Delete the details from database.

**Using collections.remove() from MongoDB**

db.users.remove({ "email": email })



The user is deleted successfully deleted from the database.

**References**

* <https://university.mongodb.com/>
* <https://docs.mongodb.com/>
* <https://www.guru99.com/mongodb-tutorials.html>
* <https://discourse.university.mongodb.com/c/M220P/6>
* <https://github.com/jkcg-learning/MFlix>