FULL STACK WITH MERN USING JAVA

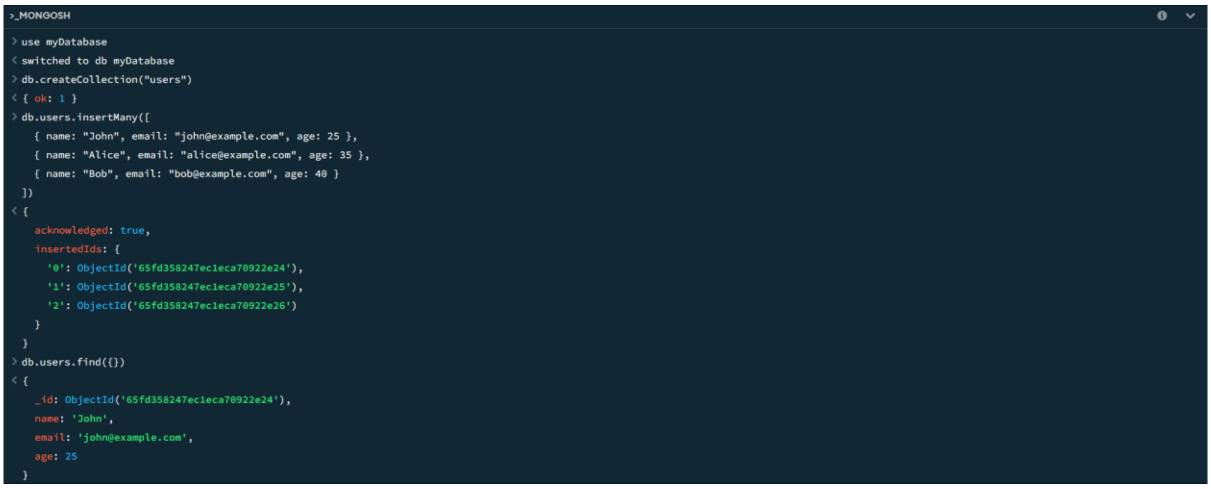
ASSIGNMENT – 4

Aim: To create databases, collections, insert documents, query data, and perform basic operations in MongoDB.

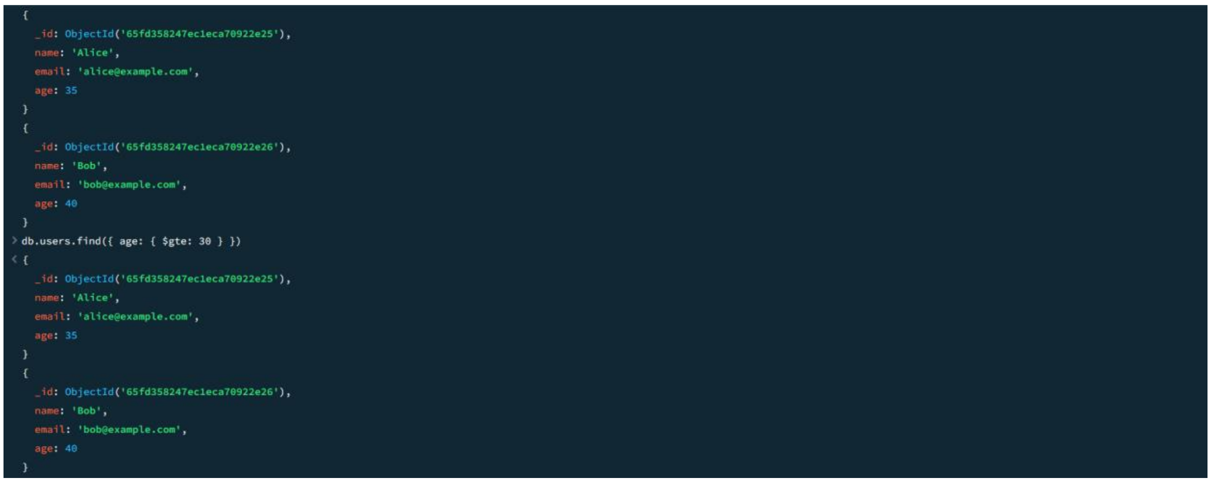
```
# Database Setup: Creating a new MongoDB database called myDatabase.

use myDatabase


# Collection Creation: Creating a collection named users within the myDatabase database.

db.createCollection("users")


# Document Insertion: Insert at least three documents into the users collection,
# each representing a user with fields such as name, email, and age.

db.users.insertMany([
  { name: "John", email: "john@example.com", age: 25 },
  { name: "Alice", email: "alice@example.com", age: 35 },
  { name: "Bob", email: "bob@example.com", age: 40 }
])


# Querying: Writing queries to retrieve:

# To get all users from the users collection.

db.users.find({})


# To find Users with an age greater than or equal to 30.

db.users.find({ age: { $gte: 30 } })
```

```
# Update Operation: Update the age of a user with a specific email address.
db.users.updateOne(
  { email: "alice@example.com" },
  { $set: { age: 36 } }
)


# Deletion Operation: Delete a user document based on a specific email address.

db.users.deleteOne({ email: "bob@example.com" })


# Index Creation: Create an index on the email field of the users collection.

db.users.createIndex({ email: 1 })
```

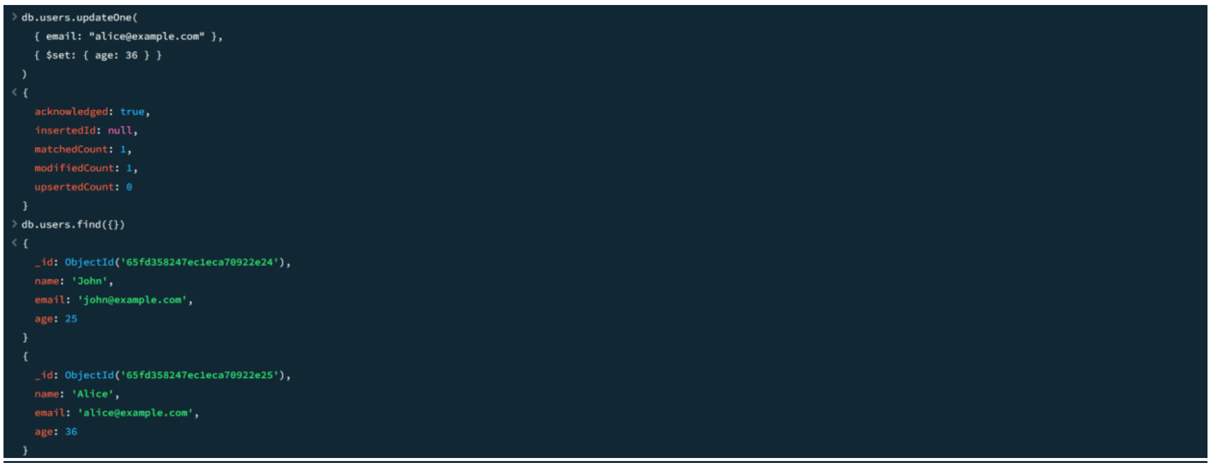
Screenshots of Mongosh



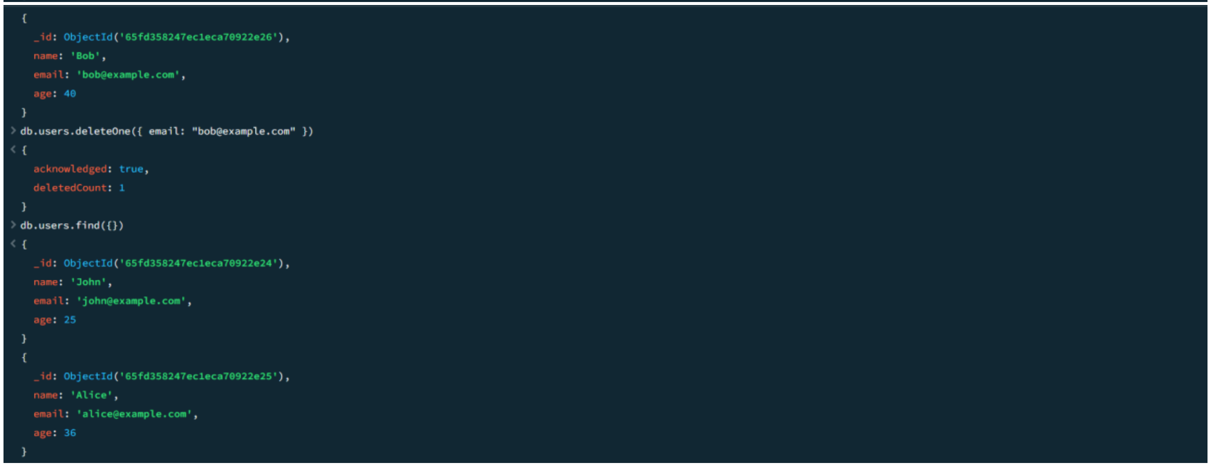
```
>_MONGOSH
> use myDatabase
< switched to db myDatabase
> db.createCollection("users")
< { ok: 1 }
> db.users.insertMany([
    { name: "John", email: "john@example.com", age: 25 },
    { name: "Alice", email: "alice@example.com", age: 35 },
    { name: "Bob", email: "bob@example.com", age: 40 }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65fd358247ec1eca70922e24'),
      '1': ObjectId('65fd358247ec1eca70922e25'),
      '2': ObjectId('65fd358247ec1eca70922e26')
    }
  }
> db.users.find({})
< {
    _id: ObjectId('65fd358247ec1eca70922e24'),
    name: 'John',
    email: 'john@example.com',
    age: 25
  }
```



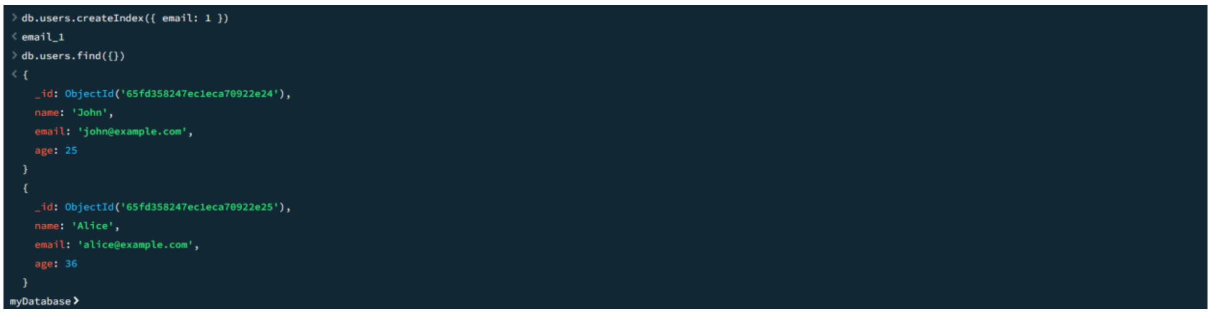
```
  {
    _id: ObjectId('65fd358247ec1eca70922e25'),
    name: 'Alice',
    email: 'alice@example.com',
    age: 35
  }
  {
    _id: ObjectId('65fd358247ec1eca70922e26'),
    name: 'Bob',
    email: 'bob@example.com',
    age: 40
  }
> db.users.find({ age: { $gte: 30 } })
< {
    _id: ObjectId('65fd358247ec1eca70922e25'),
    name: 'Alice',
    email: 'alice@example.com',
    age: 35
  }
  {
    _id: ObjectId('65fd358247ec1eca70922e26'),
    name: 'Bob',
    email: 'bob@example.com',
    age: 40
  }
```



```
> db.users.updateOne(
    { email: "alice@example.com" },
    { $set: { age: 36 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.users.find({})
< {
    _id: ObjectId('65fd358247ec1eca70922e24'),
    name: 'John',
    email: 'john@example.com',
    age: 25
  }
  {
    _id: ObjectId('65fd358247ec1eca70922e25'),
    name: 'Alice',
    email: 'alice@example.com',
    age: 36
  }
```



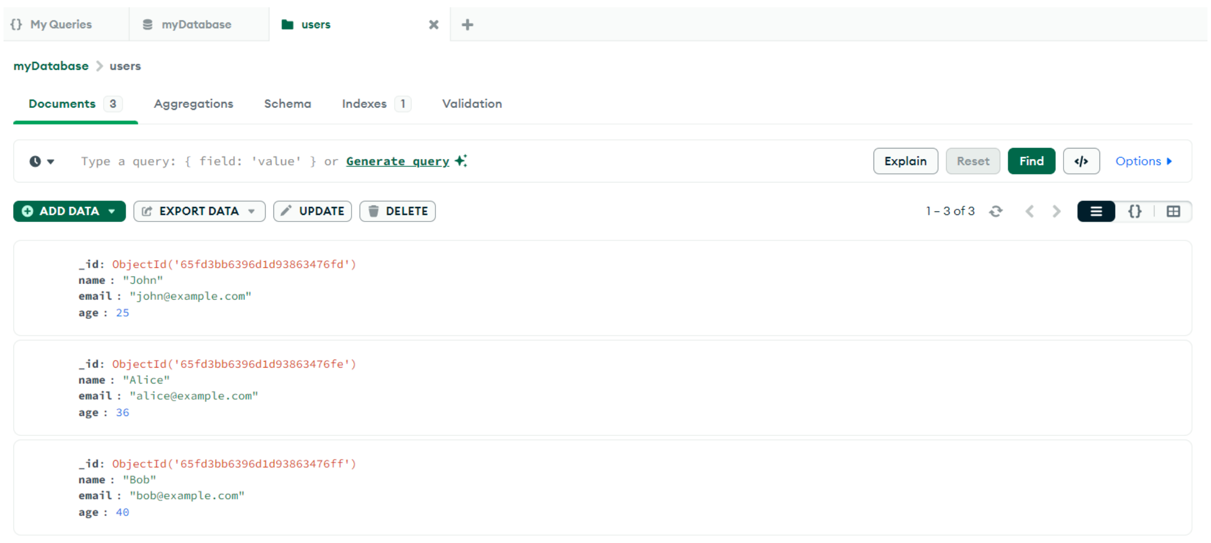
```
  {
    _id: ObjectId('65fd358247ec1eca70922e26'),
    name: 'Bob',
    email: 'bob@example.com',
    age: 40
  }
> db.users.deleteOne({ email: "bob@example.com" })
< {
    acknowledged: true,
    deletedCount: 1
  }
> db.users.find({})
< {
    _id: ObjectId('65fd358247ec1eca70922e24'),
    name: 'John',
    email: 'john@example.com',
    age: 25
  }
  {
    _id: ObjectId('65fd358247ec1eca70922e25'),
    name: 'Alice',
    email: 'alice@example.com',
    age: 36
  }
```



```
> db.users.createIndex({ email: 1 })
< email_1
> db.users.find({})
< {
    _id: ObjectId('65fd358247ec1eca70922e24'),
    name: 'John',
    email: 'john@example.com',
    age: 25
  }
  {
    _id: ObjectId('65fd358247ec1eca70922e25'),
    name: 'Alice',
    email: 'alice@example.com',
    age: 36
  }
myDatabase >
```

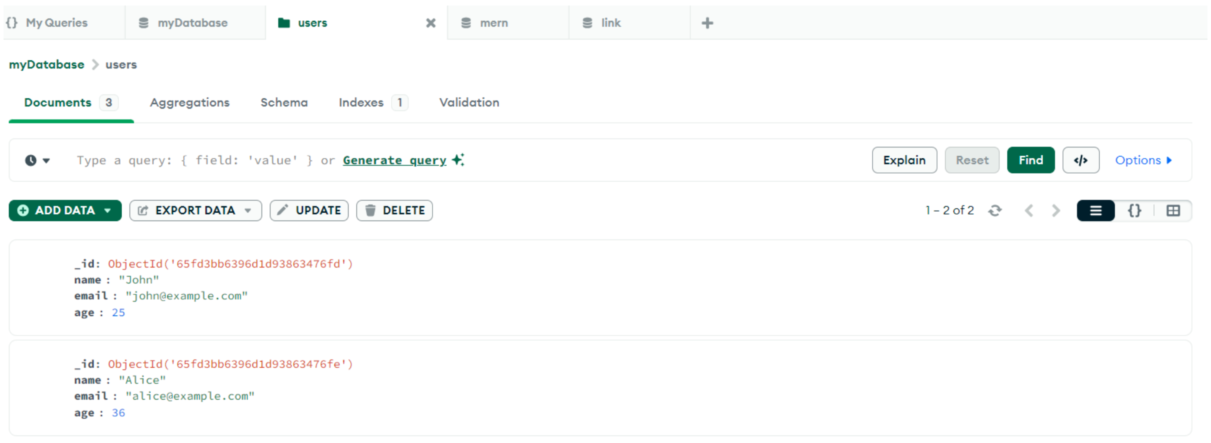
Screenshot of myDatabase documents:

Before Deletion



```
_id: ObjectId('65fd3bb6396d1d93863476fd')
name : "John"
email : "john@example.com"
age : 25

_id: ObjectId('65fd3bb6396d1d93863476fe')
name : "Alice"
email : "alice@example.com"
age : 36

_id: ObjectId('65fd3bb6396d1d93863476ff')
name : "Bob"
email : "bob@example.com"
age : 40
```

After Deletion



```
_id: ObjectId('65fd3bb6396d1d93863476fd')
name : "John"
email : "john@example.com"
age : 25

_id: ObjectId('65fd3bb6396d1d93863476fe')
name : "Alice"
email : "alice@example.com"
age : 36
```

Conclusion:

In completing this assignment, I had successfully demonstrated proficiency in MongoDB and its command-line interface, Mongosh. By following the provided guidelines, we created a MongoDB database named myDatabase and established a collection named users within it. Utilizing the document-oriented nature of MongoDB, we inserted three user documents, each containing attributes such as name, email, and age.

This understanding of MongoDB's querying capabilities was put into practice as we crafted queries to retrieve all users from the collection and specifically target users aged 30 and above.

Additionally, we executed update operations to modify user data based on their email addresses and performed deletion operations to remove user documents as needed.

Throughout the assignment, adherence to correct MongoDB commands and query operators was paramount, ensuring the accuracy of our database operations. Detailed comments were provided alongside each command, offering clarity and insight into the purpose of each step.

In conclusion, this assignment has provided invaluable hands-on experience in MongoDB database management, equipping us with the fundamental skills required to create, query, and manipulate data within a MongoDB environment using Mongosh. This practical knowledge will undoubtedly prove beneficial in future endeavors involving NoSQL databases and document-oriented data storage systems.