

# Full Stack with MERN Assignment – 4

**Name:** Adigopula Nikhitha

**Email Id :** [208x1a4201@khitguntur.ac.in](mailto:208x1a4201@khitguntur.ac.in)

**Phone :** 9398751981

**Roll Number :** 208X1A4201(AI&ML)

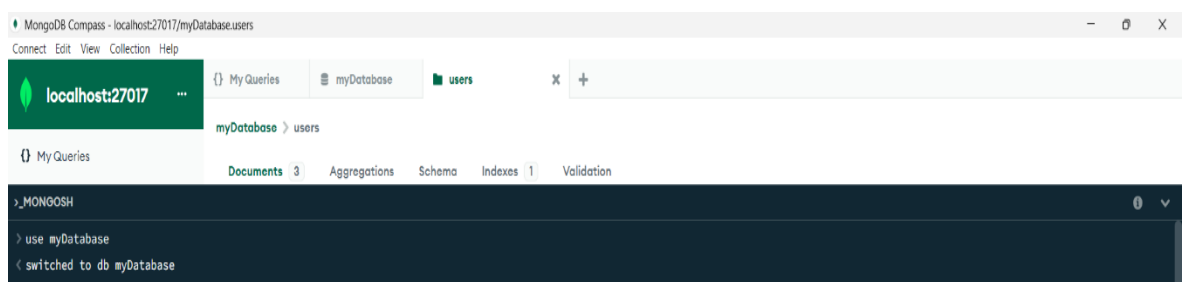
**College :** Kallam Haranadhareddy Institute Of Technology

## Creating a Database Using MongoDB and Mongosh

- **Database Setup:** First open MongoDB Compass and later on create a new MongoDB database called myDatabase.

**Command :** use myDatabase

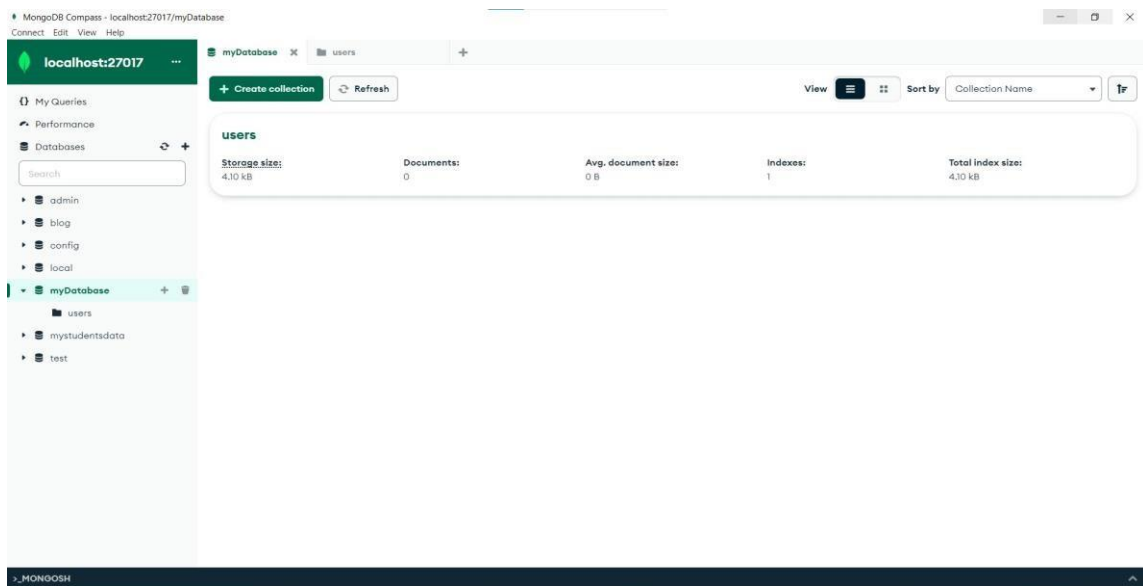
**Result:** switched to db myDatabase



- **Collection Creation:**

Creating a collection named users within the database.

**Command :** db.createCollection('users')



- **Document Insertion:** Inserting documents into the users

Collection either single or multiple , each representing a user with fields such as name, email, and age.

- **Inserting single person details:**

### **Command :**

```
db.users.insertOne({'name':'Nikki','email':'nikki@gmail.com','age':22})
```

### **Result:**

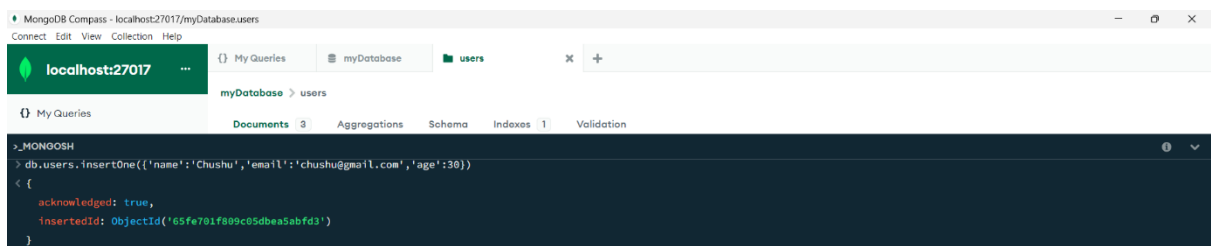
```
{  
  acknowledged: true,  
  insertedId: ObjectId('65f957fea3543e81167f43b1')  
}
```

### **Command :**

```
db.users.insertOne({'name':'Chushu','email':'chushu@gmail.com','age':30})
```

### **Result:**

```
{
  acknowledged: true,
  insertedId: ObjectId('65fe701f809c05dbea5abfd3')
}
```



- **Inserting multiple persons details:**

### **Command :**

```
db.users.insertMany([ {'name':'Maha','email':'maha@gmail.com','age':23}, {'name':'Deekshi','email':'deekshi@gmail.com','age':23} ])
```

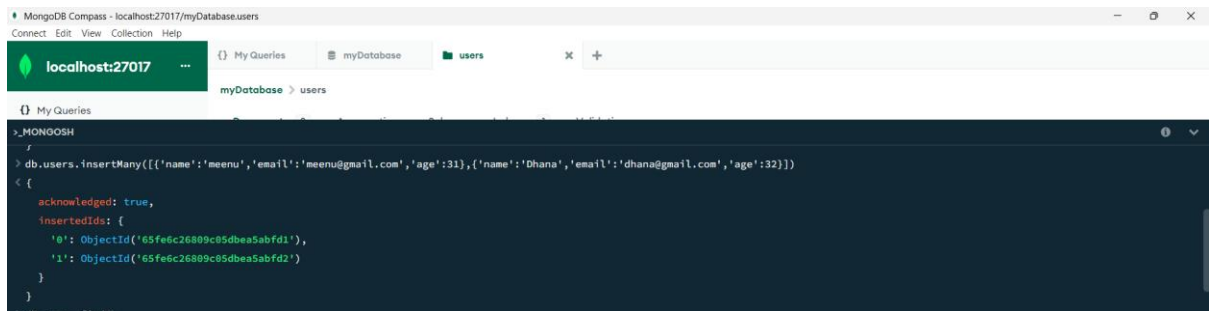
### **Result:**

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65f958b1a3543e81167f43b2'),
    '1': ObjectId('65f958b1a3543e81167f43b3')
  }
}
```

### **Command :**

```
db.users.insertMany([ {'name':'meenu','email':'meenu@gmail.com','age':31}, {'name':'Dhana','email':'dhana@gmail.com','age':32} ])
```

## Result:



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the current database is 'myDatabase'. The 'users' collection is selected. The command window shows the following command and result:

```
> db.users.insertMany([{'name':'meenu','email':'meenu@gmail.com','age':31},{'name':'Dhana','email':'dhana@gmail.com','age':32}])
```

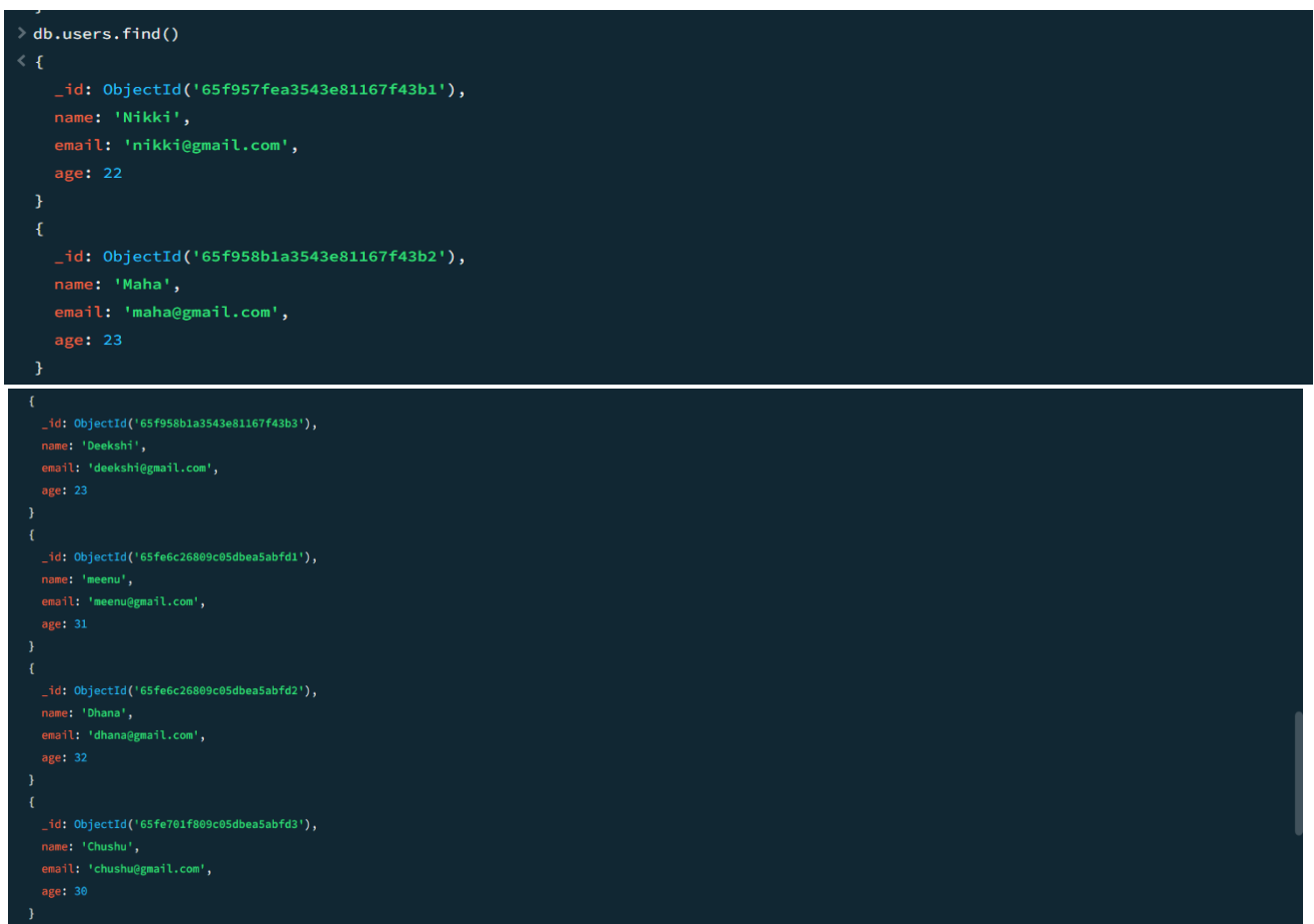
```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65fe6c26809c05d8ea5abfd1'),
    '1': ObjectId('65fe6c26809c05d8ea5abfd2')
  }
}
```

**Querying:** Writing queries to retrieve the users data.

- Finding/Retrieving all users from the users collection.

**Command :** db.users.find()

## Result:



The screenshot shows the MongoDB Compass interface. The command window shows the following command and result:

```
> db.users.find()
```

```
{
  _id: ObjectId('65f957fea3543e81167f43b1'),
  name: 'Nikki',
  email: 'nikki@gmail.com',
  age: 22
}
{
  _id: ObjectId('65f958b1a3543e81167f43b2'),
  name: 'Maha',
  email: 'maha@gmail.com',
  age: 23
}
```

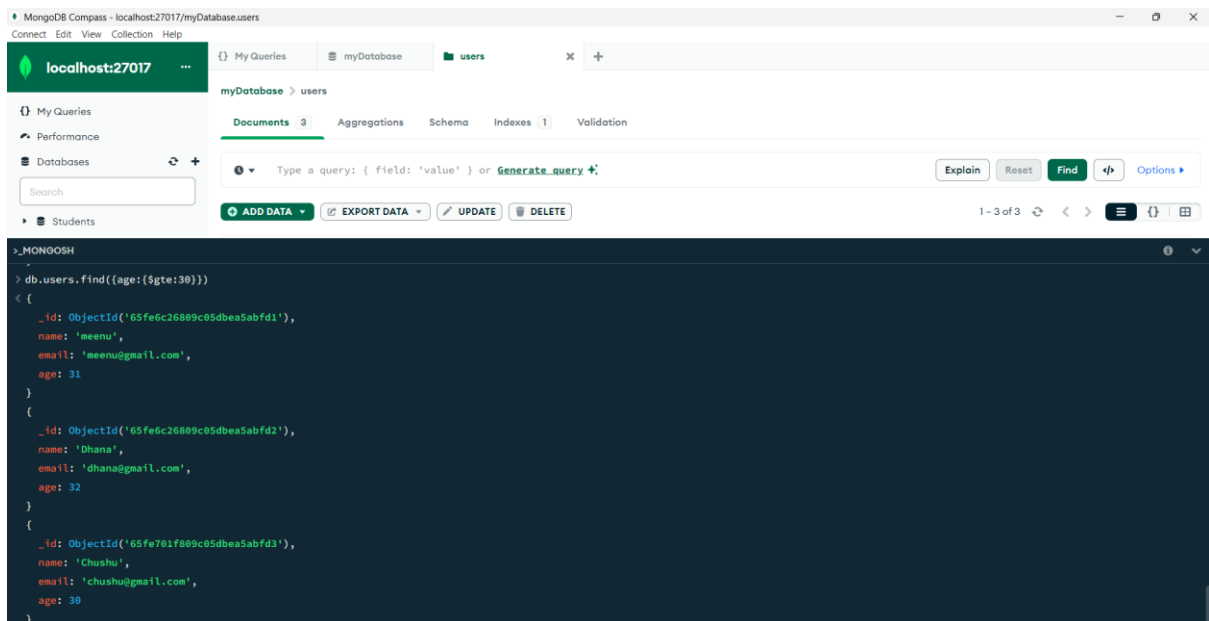
```
{
  _id: ObjectId('65f958b1a3543e81167f43b3'),
  name: 'Deekshi',
  email: 'deekshi@gmail.com',
  age: 23
}
{
  _id: ObjectId('65fe6c26809c05d8ea5abfd1'),
  name: 'meenu',
  email: 'meenu@gmail.com',
  age: 31
}
{
  _id: ObjectId('65fe6c26809c05d8ea5abfd2'),
  name: 'Dhana',
  email: 'dhana@gmail.com',
  age: 32
}
{
  _id: ObjectId('65fe701f809c05d8ea5abfd3'),
  name: 'Chushu',
  email: 'chushu@gmail.com',
  age: 30
}
```

- Retrieving Users with an age greater than or equal to 30

### Command :

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

### Result:



- **Update Operation:** Updating the age of a user (maha) with a specific email address.

### Command :

`db.users.updateOne({'email':'maha@gmail.com'},{$set: {'age':20}})`

### Result:

```
> db.users.updateOne({'email':'maha@gmail.com'},{$set:{'age':20}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

- Retrieving / Finding whether age is updated or not?

**Command :**

```
db.users.find({age:{$eq:20}})
```

**Result:**

```
> db.users.find({age:{$eq:20}})
< {
  _id: ObjectId('65f958b1a3543e81167f43b2'),
  name: 'Maha',
  email: 'maha@gmail.com',
  age: 20
}
```

- **Deletion Operation:** Deleting a user document (chushu)

based on a specific email address.

**Command :**

```
db.users.deleteOne({'email':'chushu@gmail.com'})
```

## Result:

```
> db.users.deleteOne({'email':'chushu@gmail.com'})
< {
  acknowledged: true,
  deletedCount: 1
}
```

- Finding whether age is updated or not?

## Result:

```
> db.users.find({'email':'chushu@gmail.com'})
<
> db.users.deleteOne({'email':'chushu@gmail.com'})
< {
  acknowledged: true,
  deletedCount: 0
}
myDatabase>
```

- **Index Creation:** Creating an index on the email field of the user's collection. We use db.users.createIndex() method.

**Command :** db.users.createIndex({email:1})

## Result:

```
> db.users.createIndex({ email: 1 });
< email_1
> db.users.getIndexes();
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { email: 1 }, name: 'email_1' }
]
myDatabase>
```

# Final Output :

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'myDatabase' and its collections: 'users', 'mydb', 'school', and 'test'. The 'users' collection is selected. The main panel shows the 'Documents' tab with 5 documents. Each document is a JSON object with fields: '\_id', 'name', 'email', and 'age'.

Document	_id	name	email	age
1	ObjectId('65f957fea3543e81167f43b1')	"Nikkil"	"nikk1@gmail.com"	22
2	ObjectId('65f958b1a3543e81167f43b2')	"Maha"	"maha@gmail.com"	28
3	ObjectId('65f958b1a3543e81167f43b3')	"Deekshi"	"deekshi@gmail.com"	23
4	ObjectId('65fec26889c05d8ea5abfd1')	"meenu"	"meenu@gmail.com"	31
5	ObjectId('65fec26889c05d8ea5abfd2')	"Dhana"	"dhana@gmail.com"	

The screenshot shows the MongoDB Compass interface with the 'Indexes' tab selected for the 'users' collection. It displays two indexes: '\_id' (UNIQUE) and 'email\_1' (REGULAR).

Name and Definition	Type	Size	Usage	Properties
✓ _id _id ↑	REGULAR (UNIQUE)	34.9 KB	4 (since Tue Mar 19 2024)	UNIQUE
✓ email_1 email ↑	REGULAR	20.5 KB	0 (since Sat Mar 23 2024)	