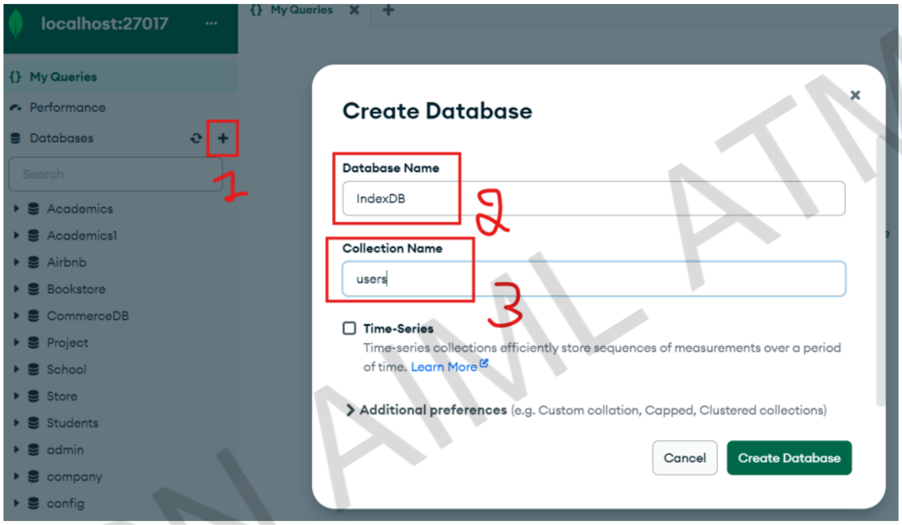
Program 8

8. a. Demonstrate creation of different types of indexes on collection (unique, sparse, compound and multikey indexes)

Create a database **IndexDB** and collection **users** in Mongo IDE.



Create Database

Database Name: IndexDB

Collection Name: users

□ Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

› Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Add the following documents in the **users collection** in MongoDB Shell.

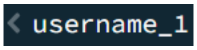
```
db.users.insert({

username: "John",

age: 30,

city:"Chennai",

"interests":["music","garden"],

Description:["good","avg","excellent"],
"hashedField": "hashedValue1",

"location": { "type": "Point", "coordinates": [ -73.97, 40.77 ] },

"createdAt": ISODate("2023-01-01T00:00:00Z") })
```

1. **Unique index:**

   A unique index ensures that the indexed field(s) do not have duplicate values

```
db.users.createIndex({ "username": 1 }, { unique: true })
```

Output:



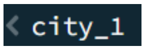
```
username_1
```

2. **Sparse Index:**

   A sparse index only includes documents that have the indexed field.

```
db.users.createIndex({ "city": 1 }, { sparse: true })
```

Output:



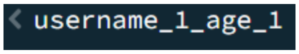
```
city_1
```

3. **Compound Index:**

   A compound index includes multiple fields within a single index.

```
db.users.createIndex({ “username”: 1, “age”: 1 })
```

Output:

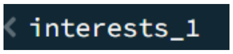


```
username_1_age_1
```

4. **Multikey Index**

   A multikey index is created on an array field, indexing each value of the array.

```
db.users.createIndex({ "interests": 1 })
```

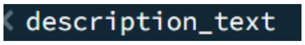


```
interests_1
```

Assuming interests is an array field in the user documents.

db.users.createIndex({ description: "text" })

Output:



```
< description_text
```

MongoDB provides text indexes to support text search queries on string content. text indexes can include any field whose value is a string or an array of string elements. Remember that you can have only one text index per collection so after creating one if you create another text index, you will get an error.

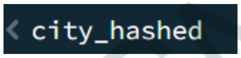
Hashed Index: Indexes where MongoDB hashes the index keys to create a more even distribution of keys.

db. users.createIndex({ city: "hashed" })

Hashed indexes are beneficial for sharding collections in MongoDB.

They distribute data across shards based on the hash value of the indexed field, improving query performance for filtering based on that field

Output:

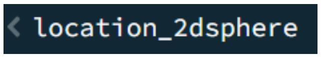


```
< city_hashed
```

Geo-spatial Index: Indexes used for geo-spatial queries.

db. users.createIndex({ location: "2dsphere" })

Output:

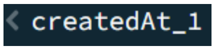


```
< location_2dsphere
```

TTL (Time-To-Live) Index: Indexes that automatically expire documents after a certain amount of time.

```
db. users.createIndex({ createdAt: 1 }, { expireAfterSeconds: 3600 })
```

Output:



```
< createdAt_1
```