

Ramprasath S  
Batch - 18 C

## WEEK 12 Activity

### **Project Title: Student Management System (Basic)**

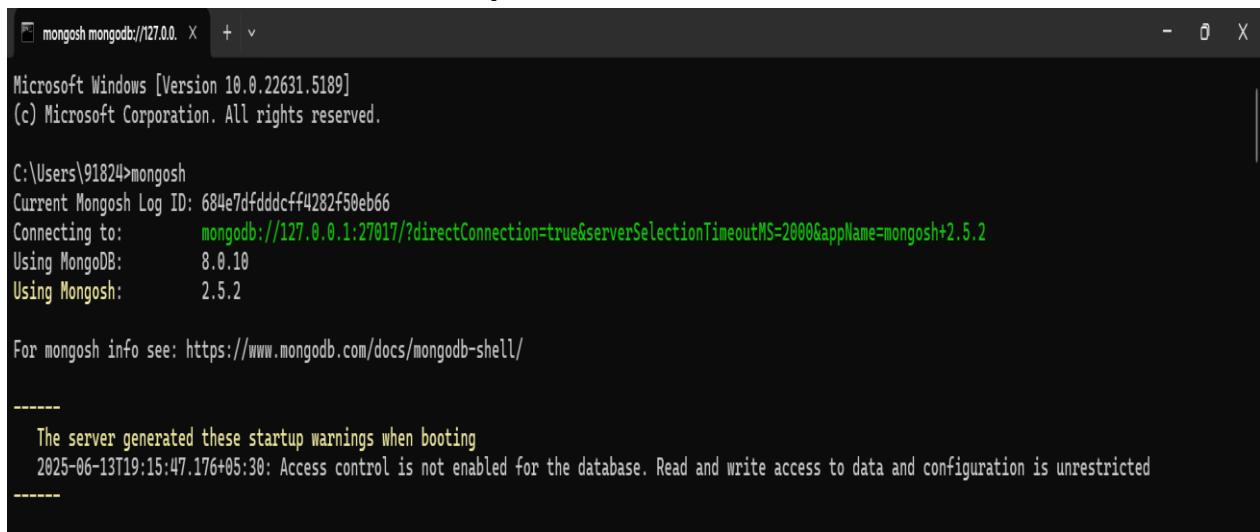
**Objective:** Build a simple database system to manage students' details using MongoDB.

**Database Name:** studentDB

**Collection Name:** students

### **Step-by-Step Tasks:**

#### **1. Database & Collection Setup**



```
mongosh mongodb://127.0.0.1:27017/ [Version 10.0.22631.5189]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91824>mongosh
Current Mongosh Log ID: 684e7dfddcff4282f50eb66
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:      8.0.18
Using Mongosh:      2.5.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-06-13T19:15:47.176+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
```

```
test> use studentDB
switched to db studentDB
studentDB> db.createCollection("students")
{ ok: 1 }
```

## 2. Insert Operations

- Insert one student:

```
studentDB> db.students.insertOne({
...   name: "Ram Prasath",
...   rollNumber: 101,
...   class: "10",
...   section: "A",
...   age: 15,
...   course: "Mathematics"
... })
...
{
  acknowledged: true,
  insertedId: ObjectId('684e7e50ddcff4282f50eb67')
}
```

### Insert multiple students:

```
studentDB> db.students.insertMany([
... {
...   name: "Sita Devi",
...   rollNumber: 102,
...   class: "10",
...   section: "B",
...   age: 14,
...   course: "Science"
... },
... {
...   name: "Arun Kumar",
...   rollNumber: 103,
...   class: "9",
...   section: "A",
...   age: 14,
...   course: "Mathematics"
... },
... {
...   name: "Meena Raj",
...   rollNumber: 104,
...   class: "8",
...   section: "C",
...   age: 13,
...   course: "English"
... }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('684e7e87ddcff4282f50eb68'),
    '1': ObjectId('684e7e87ddcff4282f50eb69'),
    '2': ObjectId('684e7e87ddcff4282f50eb6a')
  }
}
```

### 3. Read Operations

- Fetch all students:

```
studentDB> db.students.find()
[
  {
    _id: ObjectId('684e7e50ddcff4282f50eb67'),
    name: 'Ram Prasath',
    rollNumber: 101,
    class: '10',
    section: 'A',
    age: 15,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb68'),
    name: 'Sita Devi',
    rollNumber: 102,
    class: '10',
    section: 'B',
    age: 14,
    course: 'Science'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb69'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb6a'),
    name: 'Meena Raj',
    rollNumber: 104,
    class: '8',
    section: 'C',
    age: 13,
    course: 'English'
  }
]
```

**Filter by course:**

**4. Update Operations**

```
studentDB> db.students.find({ course: "Mathematics" }).pretty()
[
  {
    _id: ObjectId('684e7e50ddcff4282f50eb67'),
    name: 'Ram Prasath',
    rollNumber: 101,
    class: '10',
    section: 'A',
    age: 15,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb69'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  }
]
```

- **Update one student:**

```
studentDB> db.students.updateOne(  
...   { rollNumber: 101 },           // Filter condition  
...   { $set: { section: "B" } }    // Update operation  
... )  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
studentDB> db.students.find()  
[  
  {  
    _id: ObjectId('684e7e50ddcff4282f50eb67'),  
    name: 'Ram Prasath',  
    rollNumber: 101,  
    class: '10',  
    section: 'B',  
    age: 15,  
    course: 'Mathematics'  
  },  
  {  
    _id: ObjectId('684e7e87ddcff4282f50eb68'),  
    name: 'Sita Devi',  
    rollNumber: 102,  
    class: '10',  
    section: 'B',  
    age: 14,  
    course: 'Science'  
  },  
  {  
    _id: ObjectId('684e7e87ddcff4282f50eb69'),  
    name: 'Arun Kumar',  
    rollNumber: 103,  
    class: '9',  
    section: 'A',  
    age: 14,  
    course: 'Mathematics'  
  },  
  {  
    _id: ObjectId('684e7e87ddcff4282f50eb6a'),  
    name: 'Meena Raj',  
    rollNumber: 104,  
    class: '8',  
    section: 'C',  
    age: 13,  
    course: 'English'  
  }  
]
```

- Update many students:

```
studentDB> db.students.updateMany(  
...   {class:"10"},  
...   {$set: { course:"Science"} }  
... )  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 2,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

```
studentDB> db.students.find({class:"10"})  
[  
  {  
    _id: ObjectId('684e7e50ddcff4282f50eb67'),  
    name: 'Ram Prasath',  
    rollNumber: 101,  
    class: '10',  
    section: 'B',  
    age: 15,  
    course: 'Science'  
  },  
  {  
    _id: ObjectId('684e7e87ddcff4282f50eb68'),  
    name: 'Sita Devi',  
    rollNumber: 102,  
    class: '10',  
    section: 'B',  
    age: 14,  
    course: 'Science'  
  }  
]
```

## 5. Delete Operations

- **Delete one student:**

```
studentDB> db.students.deleteOne({ rollNumber: 101 })
{ acknowledged: true, deletedCount: 1 }
studentDB> db.students.find()
[
  {
    _id: ObjectId('684e7e87ddcff4282f50eb68'),
    name: 'Sita Devi',
    rollNumber: 102,
    class: '10',
    section: 'B',
    age: 14,
    course: 'Science'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb69'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb6a'),
    name: 'Meena Raj',
    rollNumber: 104,
    class: '8',
    section: 'C',
    age: 13,
    course: 'English'
  }
]
```

```
studentDB> db.students.deleteOne({ _id: ObjectId("684e7e87ddcff4282f50eb6a") })
{ acknowledged: true, deletedCount: 1 }
studentDB> db.students.find()
[
  {
    _id: ObjectId('684e7e87ddcff4282f50eb68'),
    name: 'Sita Devi',
    rollNumber: 102,
    class: '10',
    section: 'B',
    age: 14,
    course: 'Science'
  },
  {
    _id: ObjectId('684e7e87ddcff4282f50eb69'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  }
]
```

- Delete all :

```
studentDB> db.students.deleteMany({})
{ acknowledged: true, deletedCount: 2 }
```

## 6. Query operator practice

```
studentDB> db.students.insertMany([
...   {
...     name: "Sita Devi",
...     rollNumber: 102,
...     class: "10",
...     section: "B",
...     age: 14,
...     course: "Science"
...   },
...   {
...     name: "Arun Kumar",
...     rollNumber: 103,
...     class: "9",
...     section: "A",
...     age: 14,
...     course: "Mathematics"
...   },
...   {
...     name: "Meena Raj",
...     rollNumber: 104,
...     class: "8",
...     section: "C",
...     age: 13,
...     course: "English"
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('684e8529ddcff4282f50eb6b'),
    '1': ObjectId('684e8529ddcff4282f50eb6c'),
    '2': ObjectId('684e8529ddcff4282f50eb6d')
  }
}
```

### ◊ \$gt (Greater Than)

```
studentDB> db.students.find({ age: { $gt: 11 } })
[ {
  _id: ObjectId('684e8529ddcff4282f50eb6b'),
  name: 'Sita Devi',
  rollNumber: 102,
  class: '10',
  section: 'B',
  age: 14,
  course: 'Science'
},
{
  _id: ObjectId('684e8529ddcff4282f50eb6c'),
  name: 'Arun Kumar',
  rollNumber: 103,
  class: '9',
  section: 'A',
  age: 14,
  course: 'Mathematics'
},
{
  _id: ObjectId('684e8529ddcff4282f50eb6d'),
  name: 'Meena Raj',
  rollNumber: 104,
  class: '8',
  section: 'C',
  age: 13,
  course: 'English'
}]
```

## ◊ \$lt (Less Than)

```
studentDB> db.students.find({ age: { $lt: 15 } })  
[  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6b'),  
    name: 'Sita Devi',  
    rollNumber: 102,  
    class: '10',  
    section: 'B',  
    age: 14,  
    course: 'Science'  
  },  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6c'),  
    name: 'Arun Kumar',  
    rollNumber: 103,  
    class: '9',  
    section: 'A',  
    age: 14,  
    course: 'Mathematics'  
  },  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6d'),  
    name: 'Meena Raj',  
    rollNumber: 104,  
    class: '8',  
    section: 'C',  
    age: 13,  
    course: 'English'  
  }  
]
```

## ◊ \$gte / \$lte

```
studentDB> db.students.find({ age: { $gte: 13, $lte: 15 } })  
[  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6b'),  
    name: 'Sita Devi',  
    rollNumber: 102,  
    class: '10',  
    section: 'B',  
    age: 14,  
    course: 'Science'  
  },  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6c'),  
    name: 'Arun Kumar',  
    rollNumber: 103,  
    class: '9',  
    section: 'A',  
    age: 14,  
    course: 'Mathematics'  
  },  
  {  
    _id: ObjectId('684e8529ddcff4282f50eb6d'),  
    name: 'Meena Raj',  
    rollNumber: 104,  
    class: '8',  
    section: 'C',  
    age: 13,  
    course: 'English'  
  }  
]
```

## ◊ \$ne (Not Equal)

```
studentDB> db.students.find({ section: { $ne: "A" } })
[
  {
    _id: ObjectId('684e8529ddcff4282f50eb6b'),
    name: 'Sita Devi',
    rollNumber: 102,
    class: '10',
    section: 'B',
    age: 14,
    course: 'Science'
  },
  {
    _id: ObjectId('684e8529ddcff4282f50eb6d'),
    name: 'Meena Raj',
    rollNumber: 104,
    class: '8',
    section: 'C',
    age: 13,
    course: 'English'
  }
]
```

## Logical Operators

### ◊ \$and

```
studentDB> db.students.find({
...   $and: [
...     { class: "10" },
...     { section: "B" }
...   ]
... }).pretty()
[{
  _id: ObjectId('684e8529ddcff4282f50eb6b'),
  name: 'Sita Devi',
  rollNumber: 102,
  class: '10',
  section: 'B',
  age: 14,
  course: 'Science'
}]
```

◊ \$or

```
studentDB> db.students.find({
...   $or: [
...     { section: "A" },
...     { section: "C" }
...   ]
... }).pretty()
...
[ {
  _id: ObjectId('684e8529ddcff4282f50eb6c'),
  name: 'Arun Kumar',
  rollNumber: 103,
  class: '9',
  section: 'A',
  age: 14,
  course: 'Mathematics'
},
{
  _id: ObjectId('684e8529ddcff4282f50eb6d'),
  name: 'Meena Raj',
  rollNumber: 104,
  class: '8',
  section: 'C',
  age: 13,
  course: 'English'
}
]
```

## In/Not In Operators

◊ \$in

```
studentDB> db.students.find({ section: { $in: ["A", "B"] } }).pretty()
[
  {
    _id: ObjectId('684e8529ddcff4282f50eb6b'),
    name: 'Sita Devi',
    rollNumber: 102,
    class: '10',
    section: 'B',
    age: 14,
    course: 'Science'
  },
  {
    _id: ObjectId('684e8529ddcff4282f50eb6c'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  }
]
```

◊ \$nin

```
studentDB> db.students.find({ section: { $nin: ["B"] } }).pretty()
[
  {
    _id: ObjectId('684e8529ddcff4282f50eb6c'),
    name: 'Arun Kumar',
    rollNumber: 103,
    class: '9',
    section: 'A',
    age: 14,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('684e8529ddcff4282f50eb6d'),
    name: 'Meena Raj',
    rollNumber: 104,
    class: '8',
    section: 'C',
    age: 13,
    course: 'English'
  }
]
```

## Library System

- Create a collection called books
- Fields: title, author, genre, status (available/borrowed)
- Practice: Insert books, update status, filter by genre, and delete based on title

```
test> use library
switched to db library
library> db.books.insertMany([
...   {
...     title: "The Alchemist",
...     author: "Paulo Coelho",
...     genre: "Fiction",
...     status: "available"
...   },
...   {
...     title: "Clean Code",
...     author: "Robert C. Martin",
...     genre: "Programming",
...     status: "borrowed"
...   },
...   {
...     title: "To Kill a Mockingbird",
...     author: "Harper Lee",
...     genre: "Classic",
...     status: "available"
...   },
...   {
...     title: "1984",
...     author: "George Orwell",
...     genre: "Dystopian",
...     status: "available"
...   }
... ])
```

```
{  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId('684eb26d9981225b9750eb67'),  
    '1': ObjectId('684eb26d9981225b9750eb68'),  
    '2': ObjectId('684eb26d9981225b9750eb69'),  
    '3': ObjectId('684eb26d9981225b9750eb6a')  
  }  
}  
library> db.books.updateOne(  
...   { title: "The Alchemist" },  
...   { $set: { status: "borrowed" } }  
... )  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
library> db.books.find({ genre: "Fiction" }).pretty()  
[  
  {  
    _id: ObjectId('684eb26d9981225b9750eb67'),  
    title: 'The Alchemist',  
    author: 'Paulo Coelho',  
    genre: 'Fiction',  
    status: 'borrowed'  
  }  
]  
library> db.books.deleteOne({ title: "1984" })  
{ acknowledged: true, deletedCount: 1 }  
library> db.books.find()  
[  
  {  
    _id: ObjectId('684eb26d9981225b9750eb67'),  
    title: 'The Alchemist',  
    author: 'Paulo Coelho',  
    genre: 'Fiction',
```

```
        status: 'borrowed'
    },
    {
        _id: ObjectId('684eb26d9981225b9750eb68'),
        title: 'Clean Code',
        author: 'Robert C. Martin',
        genre: 'Programming',
        status: 'borrowed'
    },
    {
        _id: ObjectId('684eb26d9981225b9750eb69'),
        title: 'To Kill a Mockingbird',
        author: 'Harper Lee',
        genre: 'Classic',
        status: 'available'
    }
]
```