Experiment – 7: MongoDB

Name of Student	**Aryan Patankar**
Class Roll No	**D15A_33**
D.O.P.	**13/03/2025**
D.O.S.	**20/03/2025**
Sign and Grade	

1) **Aim:** To study CRUD operations in MongoDB
2) **Problem Statement:**
   A) Create a new database to storage student details of IT dept(Name, Roll no, class name) and perform the following on the database
      a) Insert one student details
      b) Insert at once multiple student details
      c) Display student for a particular class
      d) Display students of specific roll no in a class
      e) Change the roll no of a student
      f) Delete entries of particular student

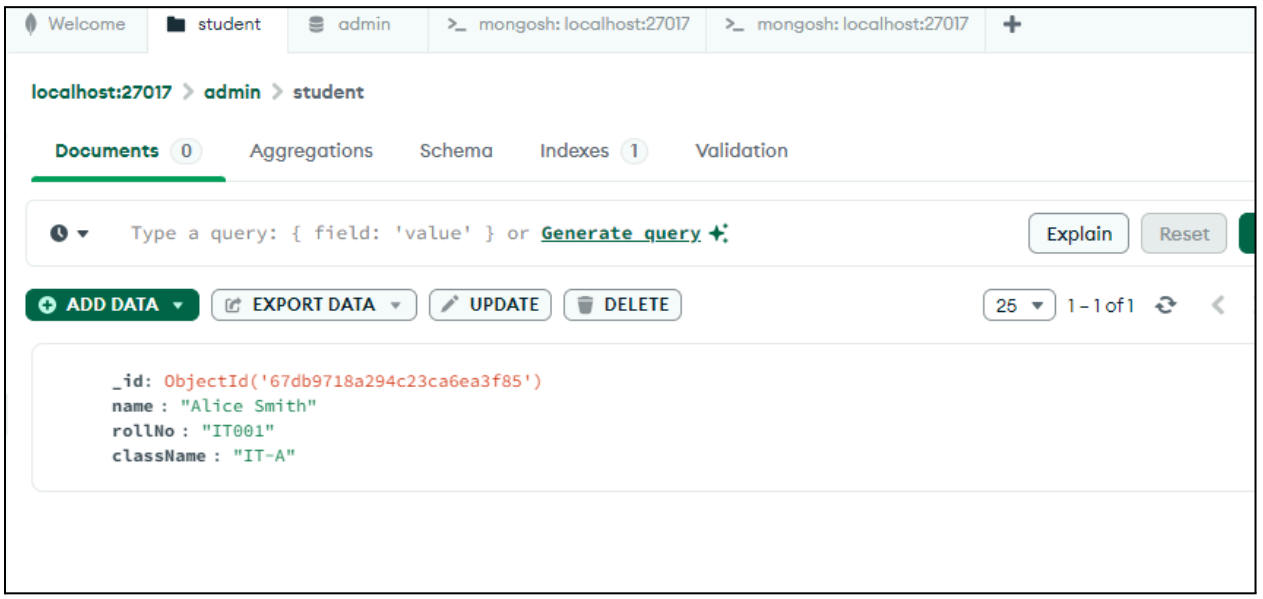
   B) Create a set of RESTful endpoints using Node.js, Express, and Mongoose for handling student data operations.
   The endpoints should support:
   ● Retrieve a list of all students.
   ● Retrieve details of an individual student by ID.
   ● Add a new student to the database.
   ● Update details of an existing student by ID.
   ● Delete a student from the database by ID.
   Connect the server to MongoDB using Mongoose, and store student data with attributes: name, age, and grade.

3) **Output:**

A)

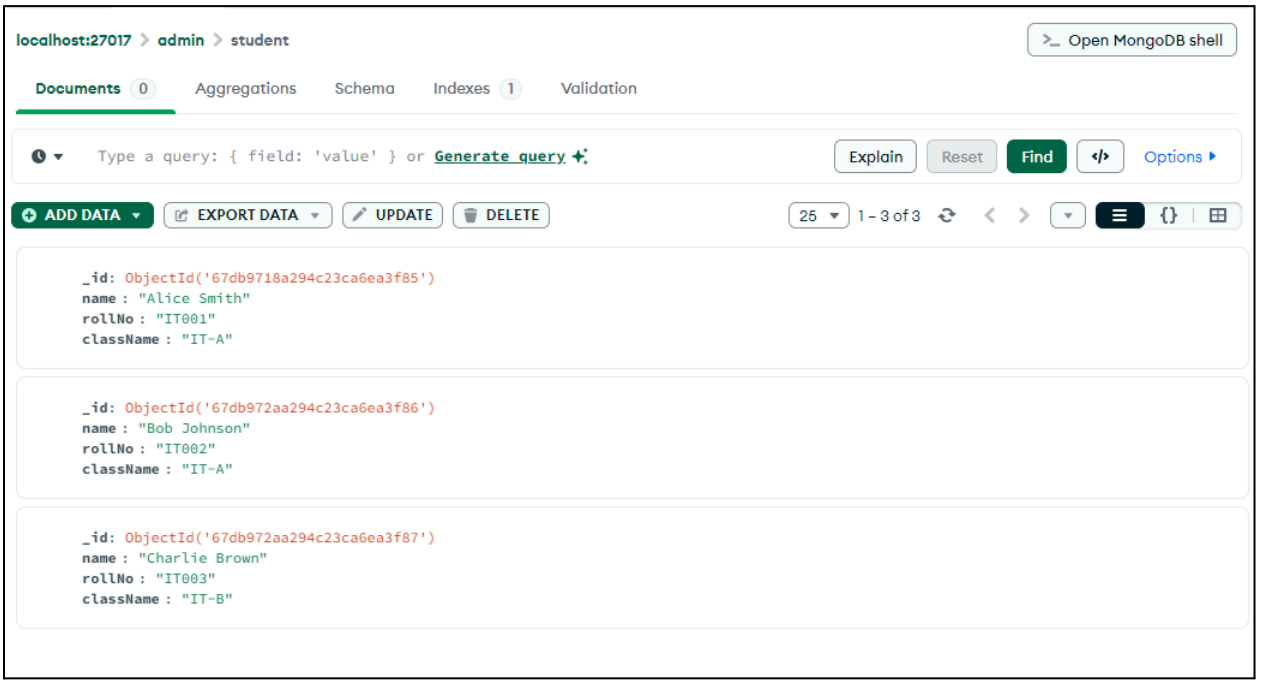
1) Insert one student details

```
> db.student.insertOne({
    name: "Alice Smith",
    rollNo: "IT001",
    className: "IT-A"
  })
< {
    acknowledged: true,
    insertedId: ObjectId('67db9718a294c23ca6ea3f85')
  }
```



2) Insert at once multiple student details

```
> db.student.insertMany([
    { name: "Bob Johnson", rollNo: "IT002", className: "IT-A" },
    { name: "Charlie Brown", rollNo: "IT003", className: "IT-B" }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('67db972aa294c23ca6ea3f86'),
      '1': ObjectId('67db972aa294c23ca6ea3f87')
    }
  }
```



3)Display students by roll number in a particular class

```
> db.student.find({ className: "IT-A" })
< {
    _id: ObjectId('67db9718a294c23ca6ea3f85'),
    name: 'Alice Smith',
    rollNo: 'IT001',
    className: 'IT-A'
  }
  {
    _id: ObjectId('67db972aa294c23ca6ea3f86'),
    name: 'Bob Johnson',
    rollNo: 'IT002',
    className: 'IT-A'
  }
```

4) Display students of specific roll no in a class

```
> db.student.find({ rollNo: "IT002", className: "IT-A" })
< {
    _id: ObjectId('67db972aa294c23ca6ea3f86'),
    name: 'Bob Johnson',
    rollNo: 'IT002',
    className: 'IT-A'
  }
```

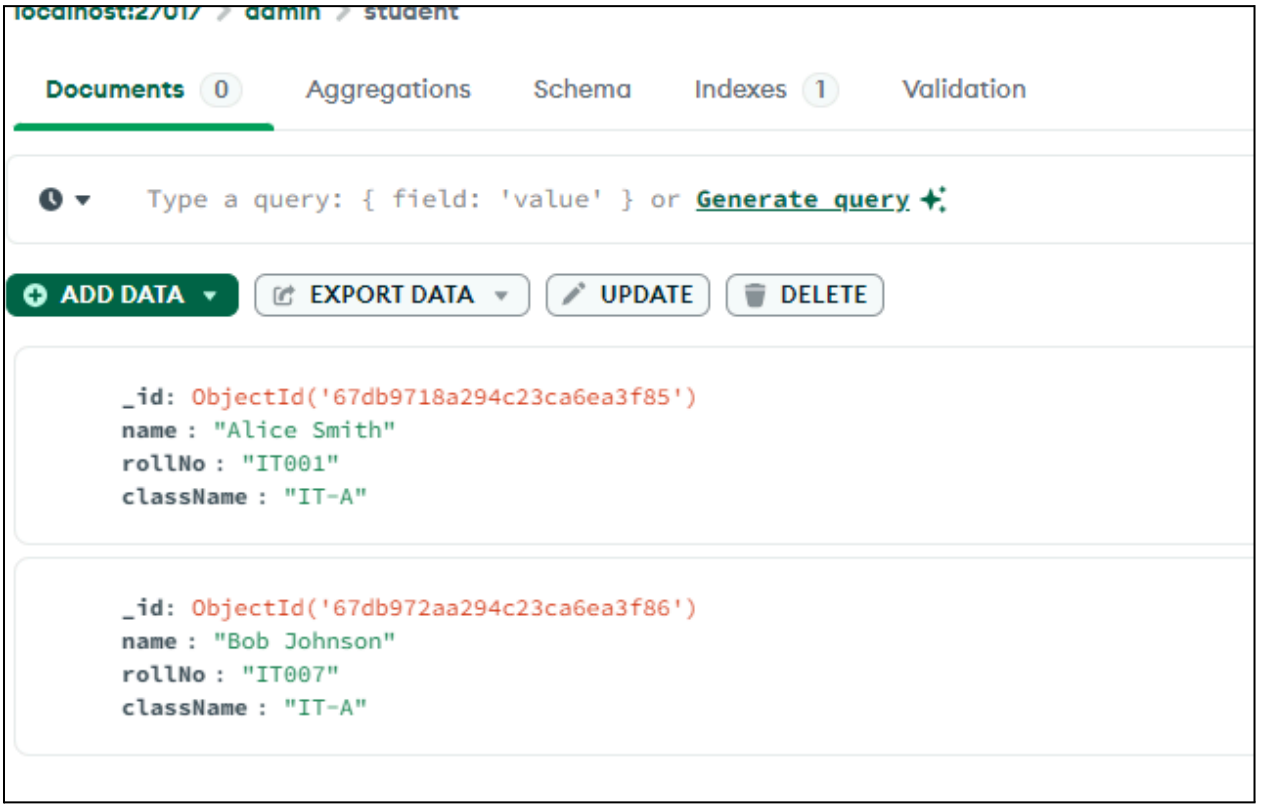
5) Change the roll no of a student

```
> db.student.updateOne(
    { name: "Bob Johnson" },
    { $set: { rollNo: "IT007" } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

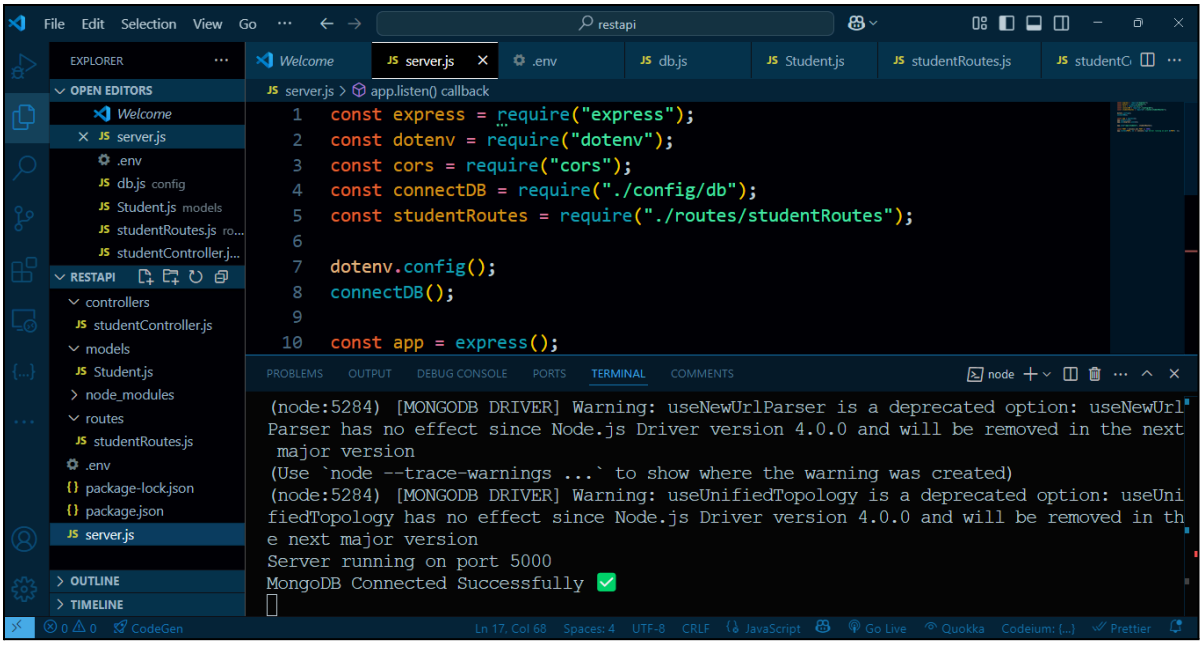
```
_id: ObjectId('67db972aa294c23ca6ea3f86')
name : "Bob Johnson"
rollNo : "IT007"
className : "IT-A"
```

6) Delete entries of particular student

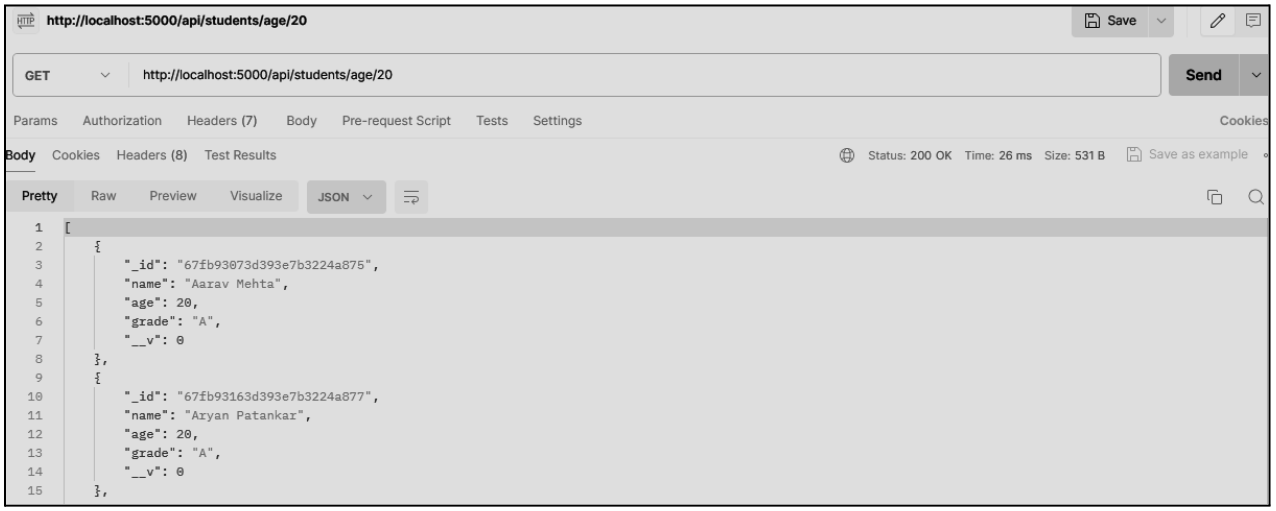
```
> db.student.deleteOne({ rollNo: "IT003" })
< {
    acknowledged: true,
    deletedCount: 1
  }
```



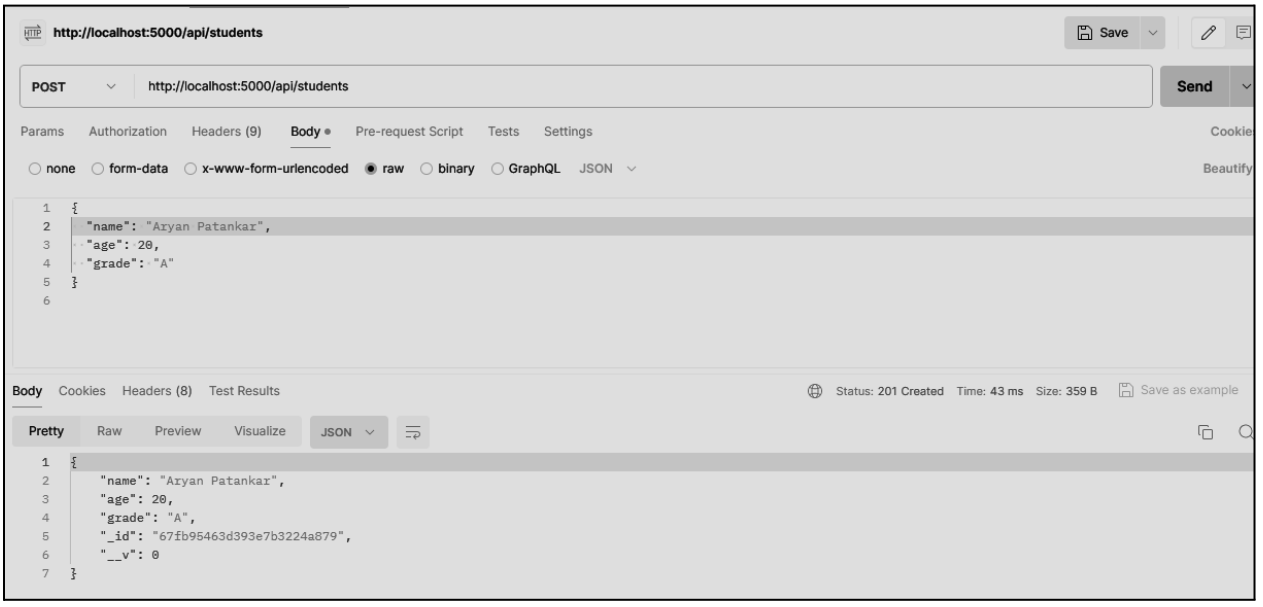
B)**Restful API's**



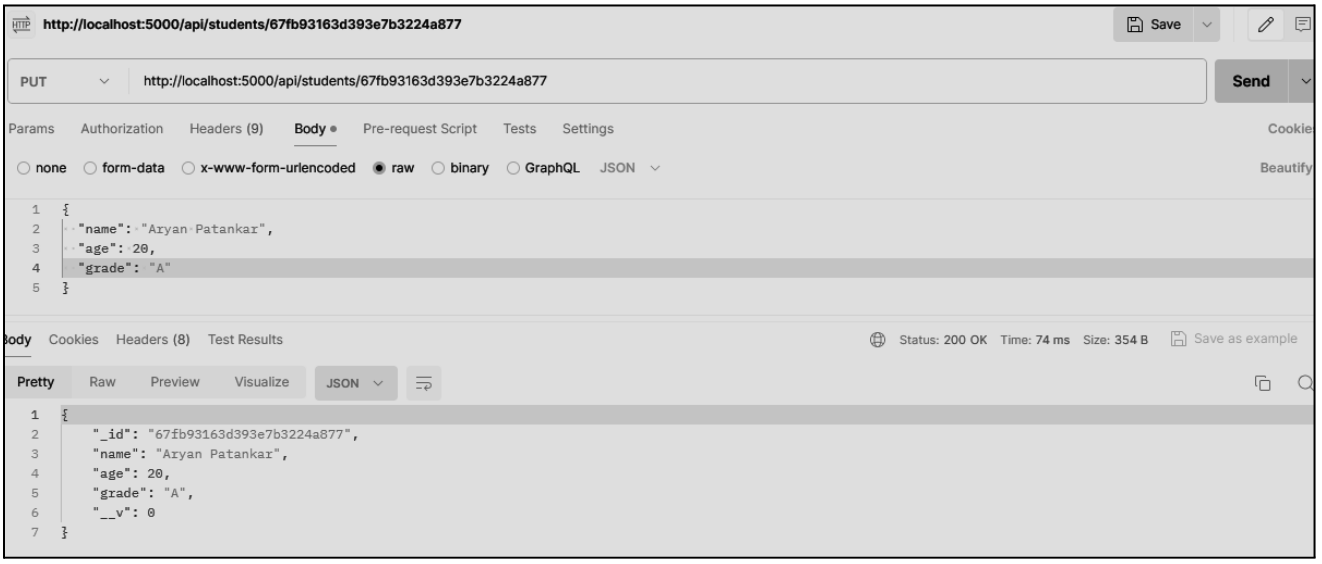
1)Retrieve details of an individual student by ID



2)Add a new student to the database.



3)Update details of an existing student by ID.



4)Delete a student from the database by ID