EXPERIMENT NO. 7 - MongoDB

Name of Student	**Bhagyesh Patil**
Class Roll No	**D15A_35**
D.O.P.	**13/02/2025**
D.O.S.	**20/03/2025**
Sign and Grade	

AIM: To study CRUD operations in MongoDB

PROBLEM STATEMENT:

1. 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

2. Create a set of RESTful endpoints using Node.js, Express, and Mongoose for handling student data operations. The endpoints should support:
    a. Retrieve a list of all students.
    b. Retrieve details of an individual student by ID.
    c. Add a new student to the database.
    d. Update details of an existing student by ID.
    e. 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.

OUTPUT:

Step 1: Use or Create a Database and Create and Use a Collection (e.g., students)

use IT_Dept_Students

db.createCollection("students")

```
>_MONGOSH
> use IT_Dept_Students
< switched to db IT_Dept_Students
> db.createCollection("students")
< { ok: 1 }
```

Step 2:

a) **Insert One Student Detail**

```
> db.students.insertOne({
    name: " Bhagyesh patil ",
    roll_no: 35,
    class_name: "IT-D15A"
})
< {
  acknowledged: true,
  insertedId: ObjectId('67fd01f908aa5808a6f9fec8')
}
```

b) Insert Multiple Student Details at Once

```
> db.students.insertMany([
    { name: "Kartik Bhatt", roll_no: 03, class_name: "IT-D15A" },
    { name: "Prajjwal Pandey", roll_no: 32  , class_name: "IT-D15A" },
    { name: "Aryan Dangat", roll_no: 12, class_name: "IT-D15A" },
    {name: "Swaraj Patil", roll_no: 39, class_name: "IT-D15A"}
])
```

c) **Display Students of a Particular Class:**

```
> db.students.find({ class_name: "IT-D15A" })
< {
    _id: ObjectId('67fcffbe08aa5808a6f9fec4'),
    name: 'Kartik Bhatt',
    roll_no: 3,
    class_name: 'IT-D15A'
  }
  {
    _id: ObjectId('67fcffbe08aa5808a6f9fec5'),
    name: 'Prajjwal Pandey',
    roll_no: 32,
    class_name: 'IT-D15A'
  }
  {
    _id: ObjectId('67fcffbe08aa5808a6f9fec6'),
    name: 'Aryan Dangat',
    roll_no: 12,
    class_name: 'IT-D15A'
  }
  {
    _id: ObjectId('67fcffbe08aa5808a6f9fec7'),
    name: 'Swaraj Patil',
    roll_no: 39,
    class_name: 'IT-D15A'
  }
```

d) **Display Students of a Specific Roll No in a Class**

```
> db.students.find({ class_name: "IT-D15A", roll_no: 25 })
< {
    _id: ObjectId('67fd01f908aa5808a6f9fec8'),
    name: 'Bhagyesh Patil',
    roll_no: 25,
    class_name: 'IT-D15A'
  }
```

e) Change the Roll No of a Student

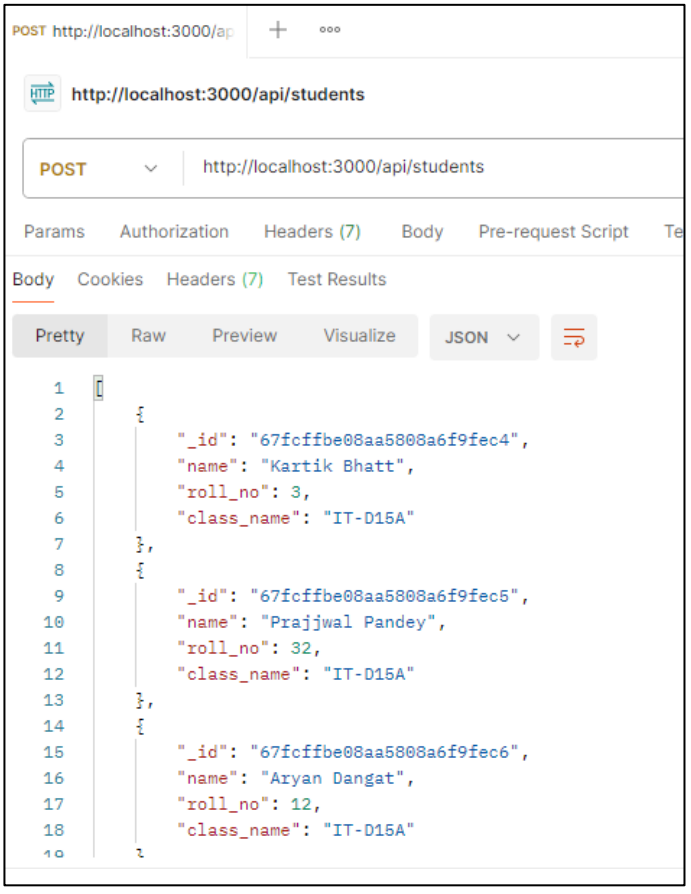
```
> db.students.updateOne(
      { name: "Bhagyesh Patil" },   // Filter
      { $set: { roll_no: 0035 } }       // Update
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

f) Delete Entry of a Particular Student

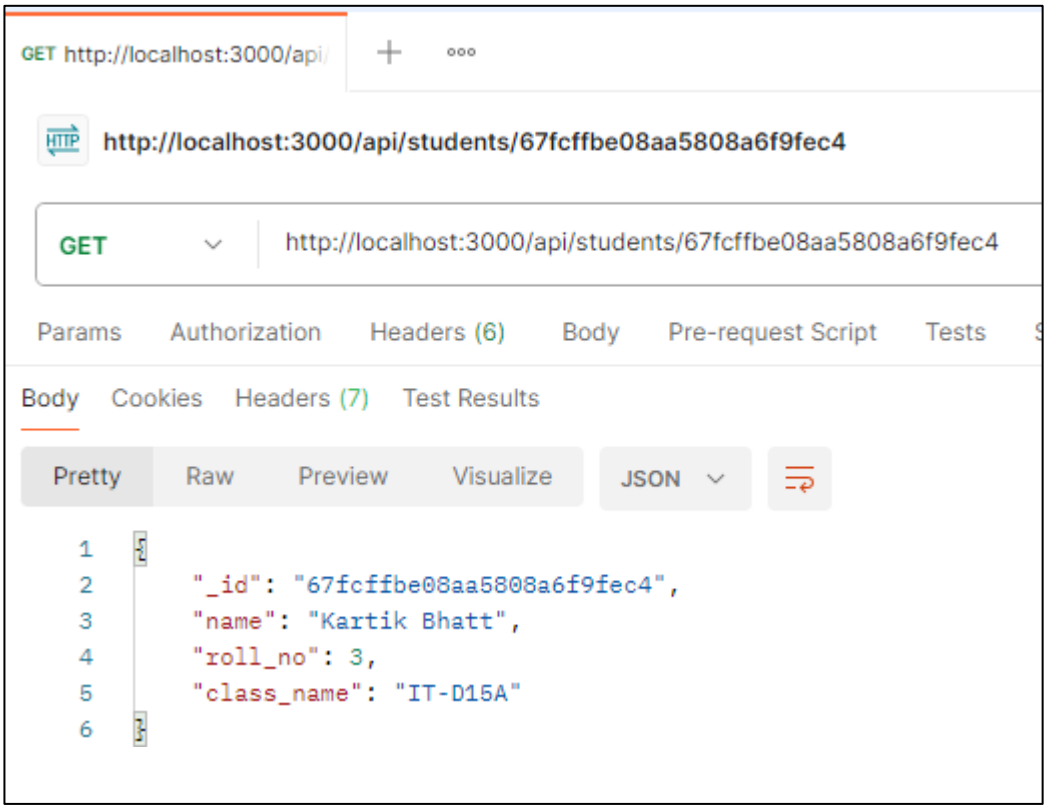
```
> db.students.deleteOne({ name: "Swaraj Patil" })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

Restful API:

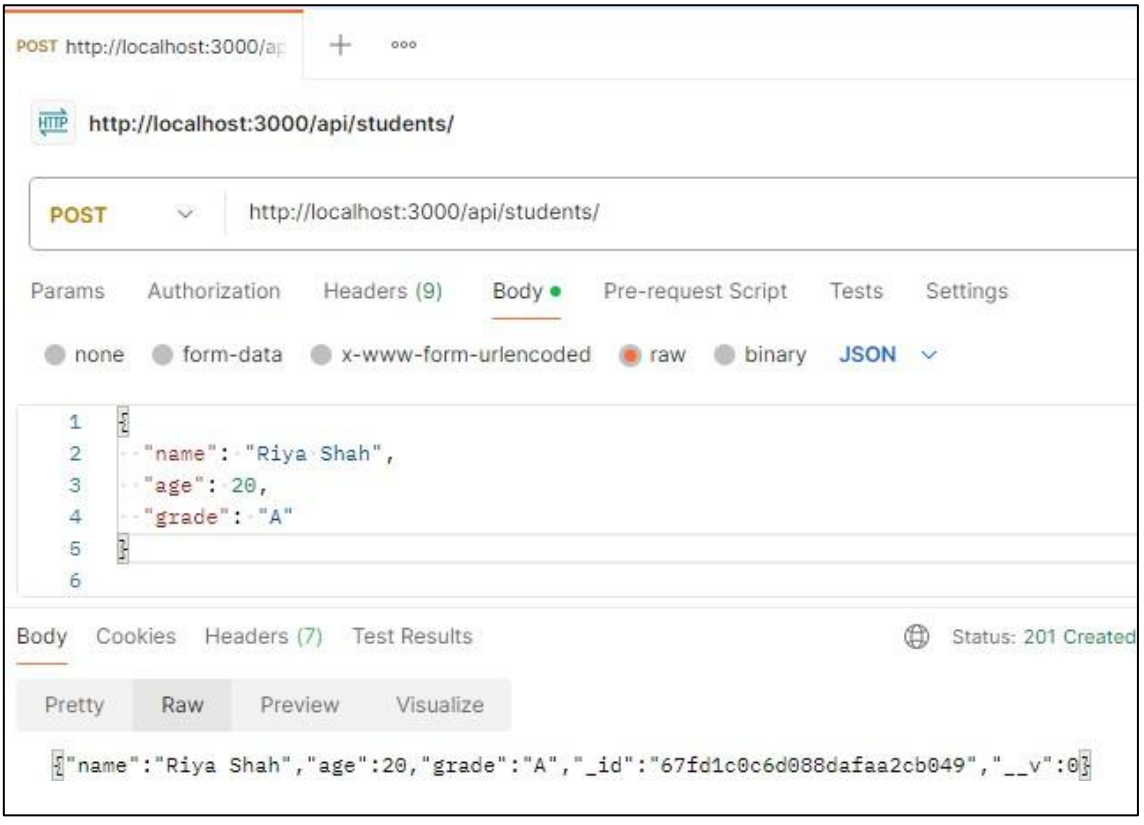
a. Retrieve a list of all students.



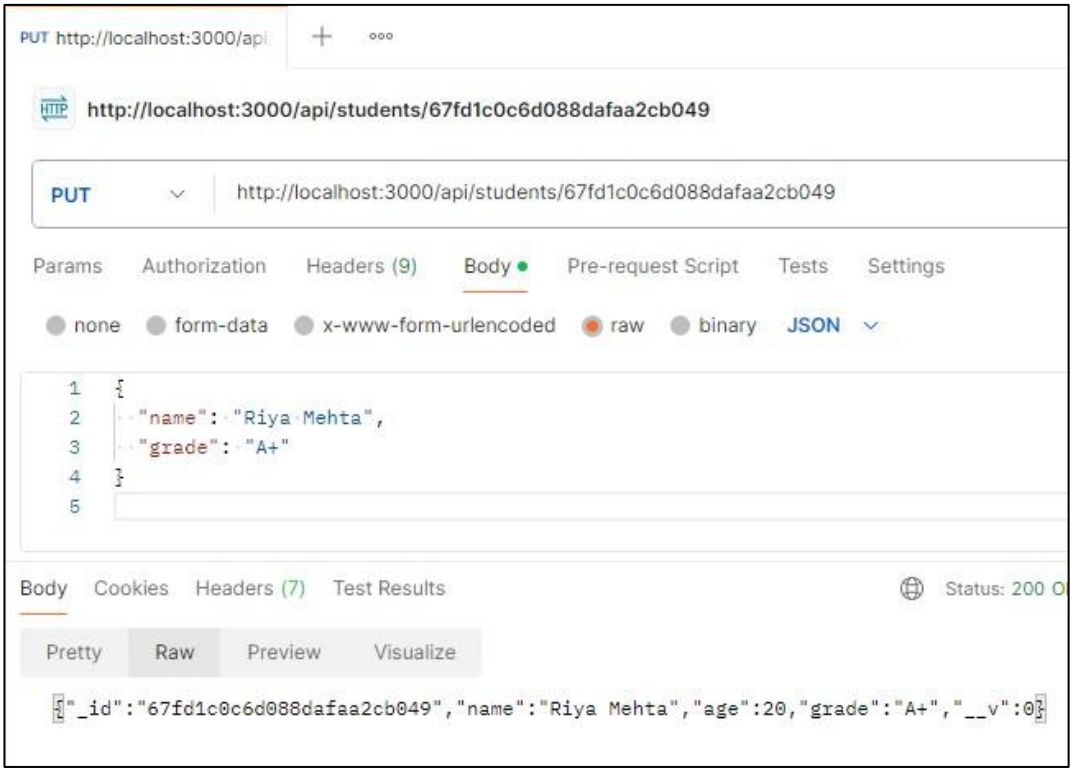
b. Retrieve details of an individual student by ID.



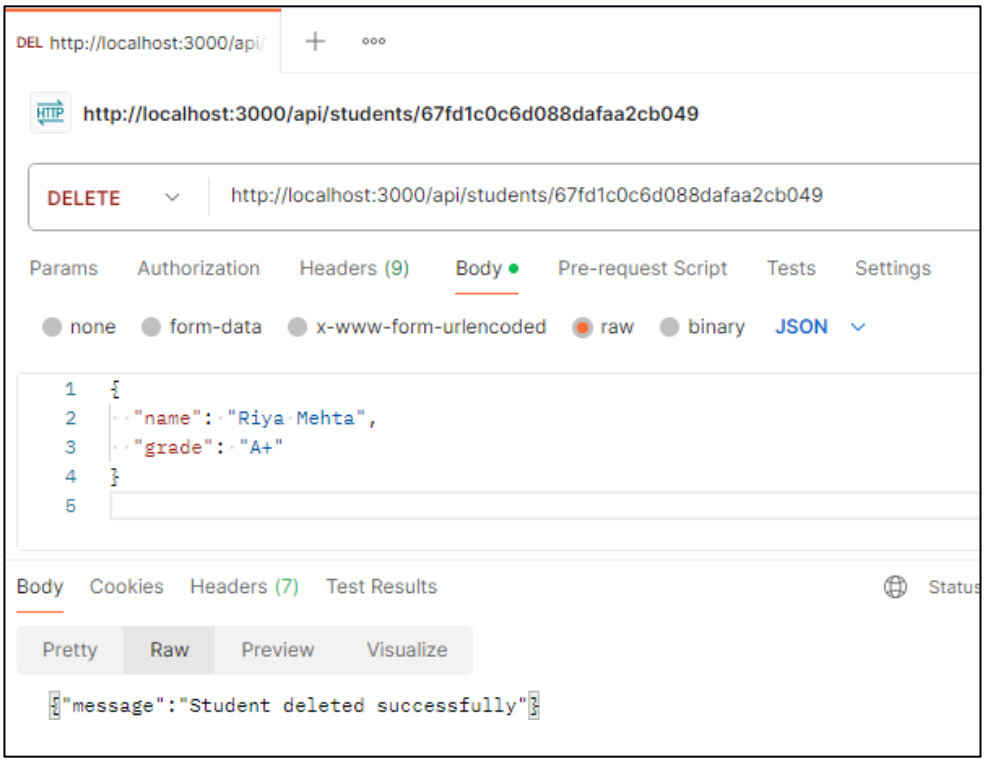
c. Add a new student to the database.



d. Update details of an existing student by ID.



e. Delete a student from the database by ID.



CONCLUSION

In this experiment, we successfully performed CRUD operations in **MongoDB** and implemented a **RESTful API** using **Node.js, Express, and Mongoose**. We learned how to create, read, update, and delete student records both via **MongoDB shell commands** and **API endpoints**.