EXPERIMENT NO. 7 - MongoDB

Name of Student	Rohan Lalchandani
Class Roll No	D15A_25
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.

GITHUB LINK:

https://github.com/Rohan-Lalchandani08/WebX Lab/tree/main/WebX%20Exp%207

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 }</pre>
```

Step 2:

a) Insert One Student Detail

```
> db.students.insertOne({
          name: "Rohan Lalchandani",
          roll_no: 25,
          class_name: "IT-D15A"
})

{ {
         acknowledged: true,
         insertedId: ObjectId('67fd01f908aa5808a6f9fec8')
}
```

b) Insert Multiple Student Details at Once

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: 'Rohan Lalchandani',
    roll_no: 25,
    class_name: 'IT-D15A'
}</pre>
```

e) Change the Roll No of a Student

f) Delete Entry of a Particular Student

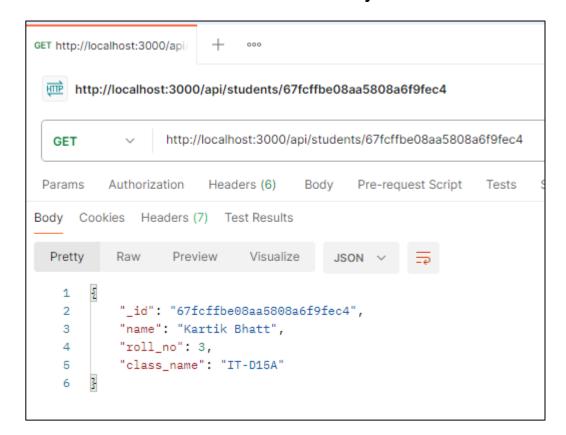
```
> db.students.deleteOne({ name: "Swaraj Patil" })
< {
    acknowledged: true,
    deletedCount: 1
}</pre>
```

Restful API:

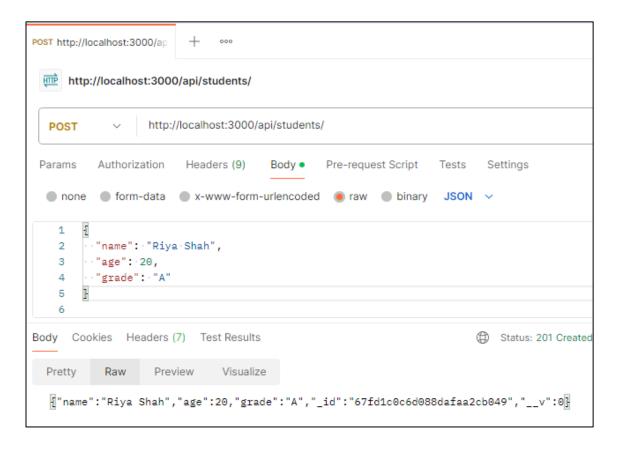
a. Retrieve a list of all students.

```
POST http://localhost:3000/ap +
 http://localhost:3000/api/students
  POST v http://localhost:3000/api/students
Params Authorization Headers (7) Body Pre-request Script
Body Cookies Headers (7) Test Results
 Pretty Raw Preview Visualize JSON V
              "_id": "67fcffbe08aa5808a6f9fec4",
              "name": "Kartik Bhatt",
   4
              "roll_no": 3,
              "class_name": "IT-D15A"
   8
              "_id": "67fcffbe08aa5808a6f9fec5",
   9
             "name": "Prajjwal Pandey",
  10
  11
              "roll_no": 32,
              "class_name": "IT-D15A"
  12
  13
  14
               "_id": "67fcffbe08aa5808a6f9fec6",
  15
              "name": "Aryan Dangat",
  16
              "roll_no": 12,
  17
  18
              "class_name": "IT-D15A"
  10
```

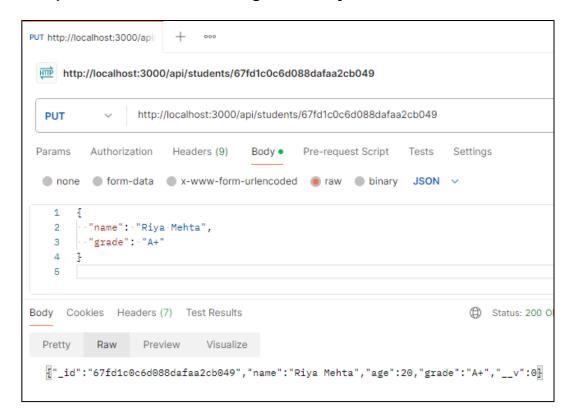
b. Retrieve details of an individual student by ID.



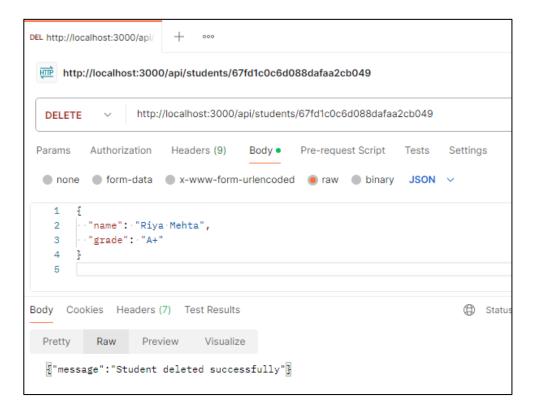
c. Add a new student to the database.



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**.