WEBX EXP7

Name of the Student	**Ria Chaudhari**
Class and Roll No	**D15A07**
DOP	
DOS	
Sign and Grade	

AIM: To study CRUD operations in MongoDB

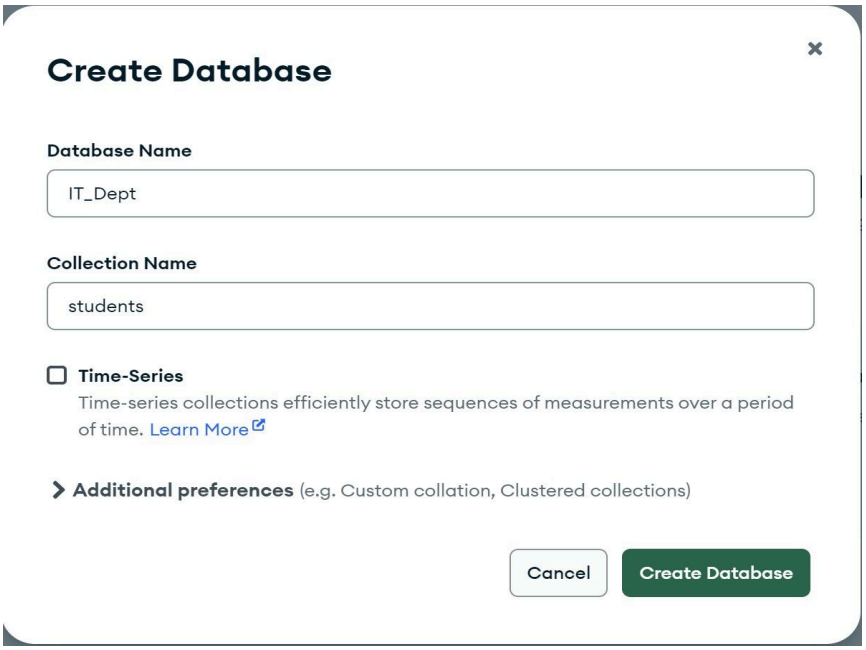
PROBLEM STATEMENT:
1. Create a new database to storage student details of IT dept(Name, Roll no, class name) and perform the following on the database
a. Insert one student details
b. Insert at once multiple student details
c. Display student for a particular class
d. Display students of specific roll 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.

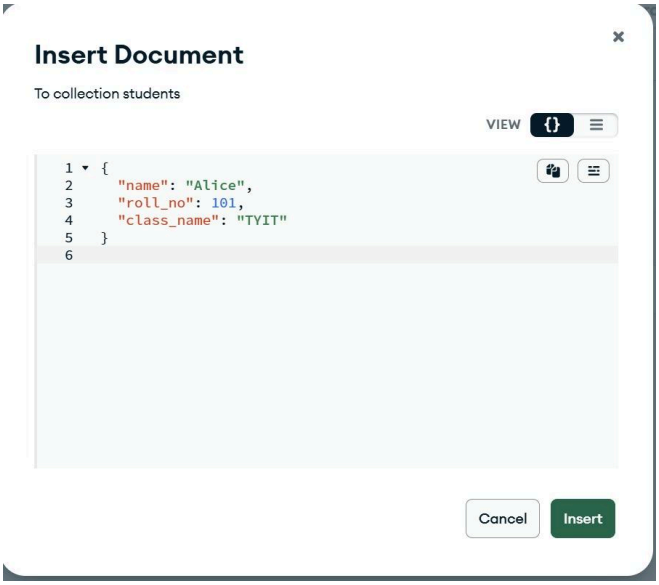
GITHUB LINK: https://github.com/riachaudhari/webx-exp7

OUTPUT:

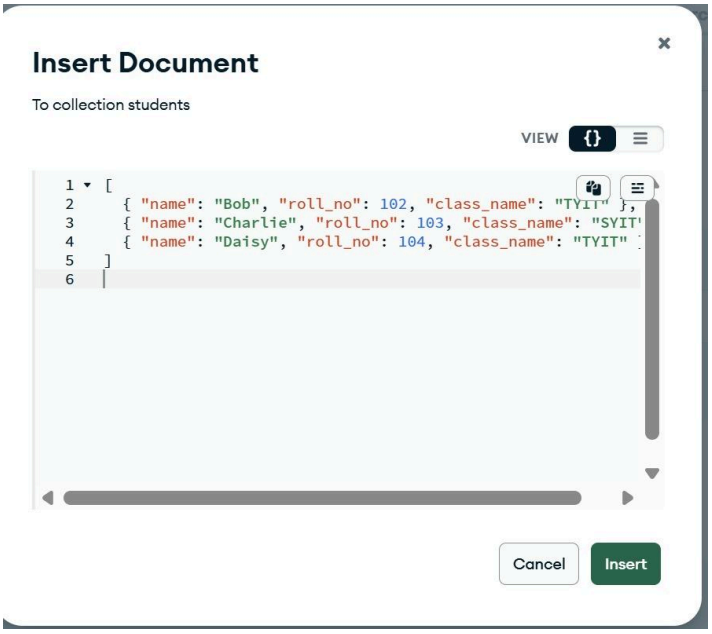
1. Create a database to store student details of IT Department



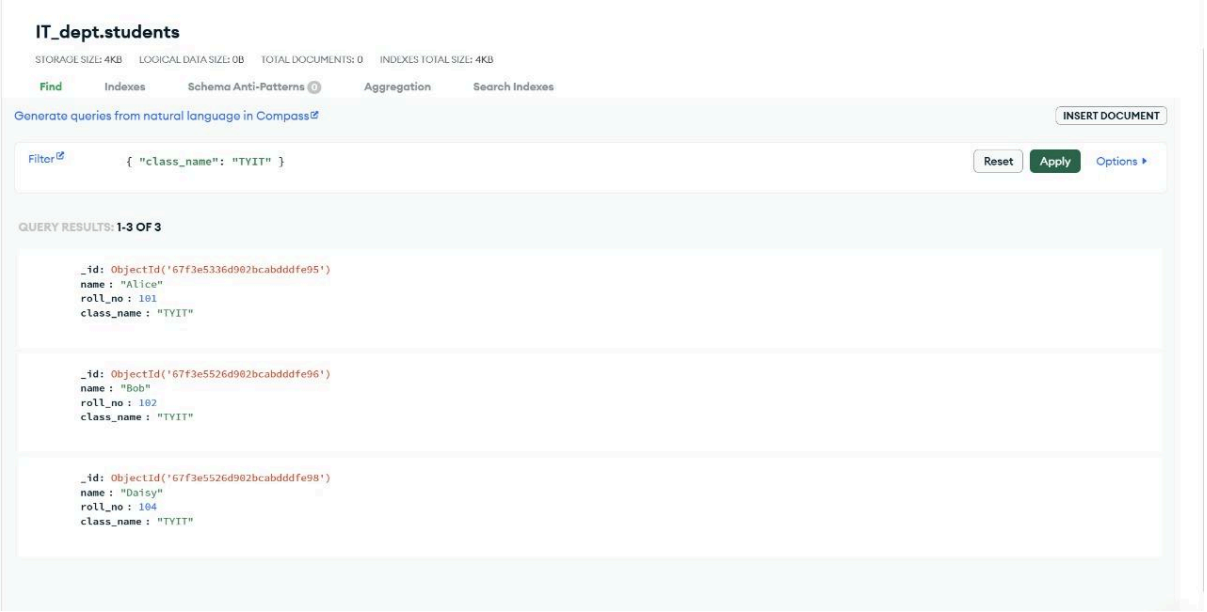
a) Insert one Student Detail



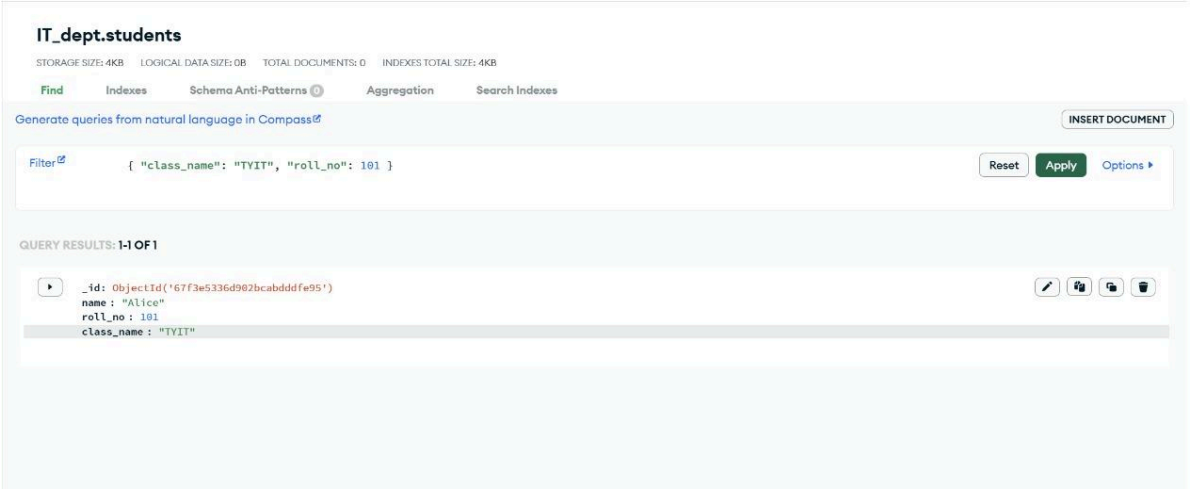
b)Insert multiple Student Details



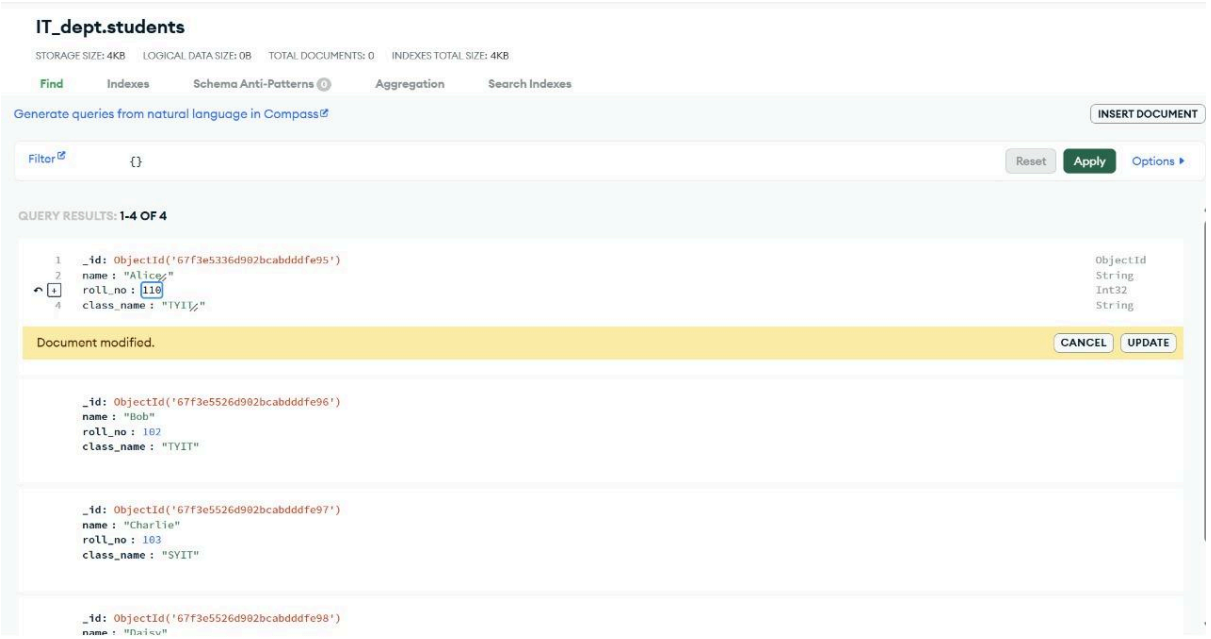
Display Students for a Particular Class
{ "className": "TYIT" }



d) Display Student of a Specific Roll Number in a Class
{ "className": "TYIT", "rollNo": 101 }

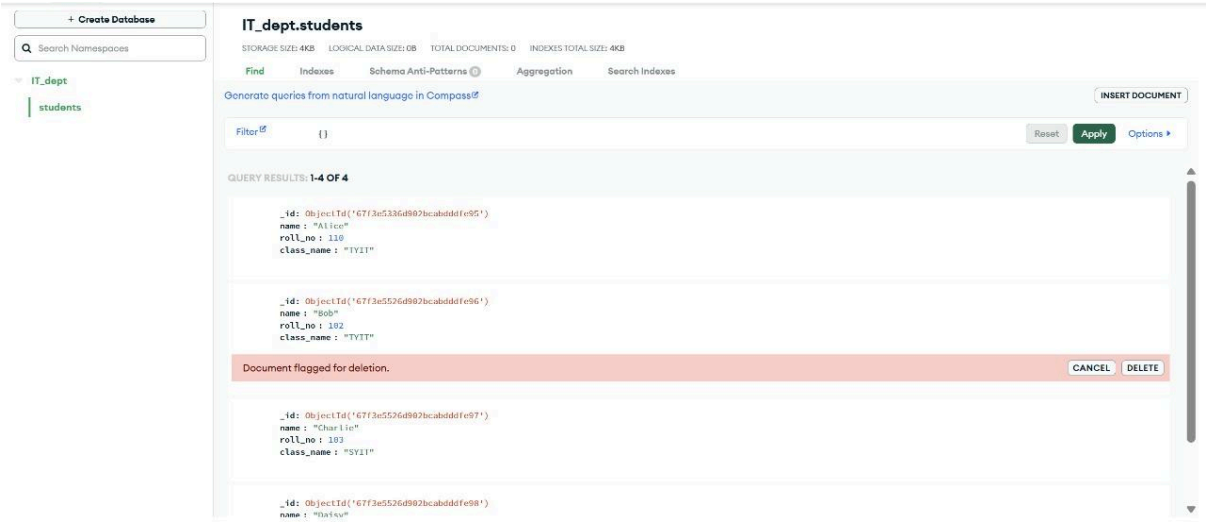


e) Change the Roll Number of a Student Filter:
{ "name": "Alice" }
Update:

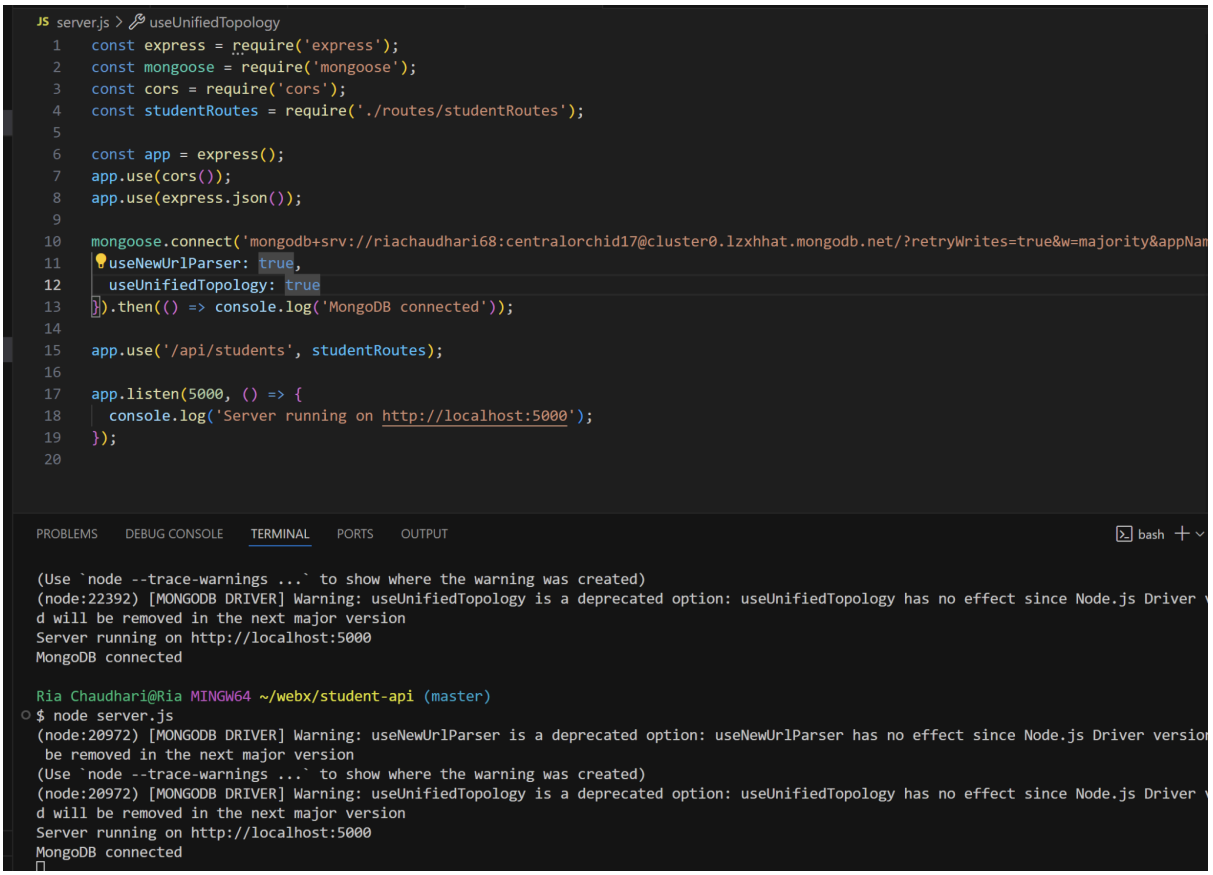


f) Delete Entries of a Particular Student Filter:
{ "name": "Bob" }

Delete:



2. Restful api



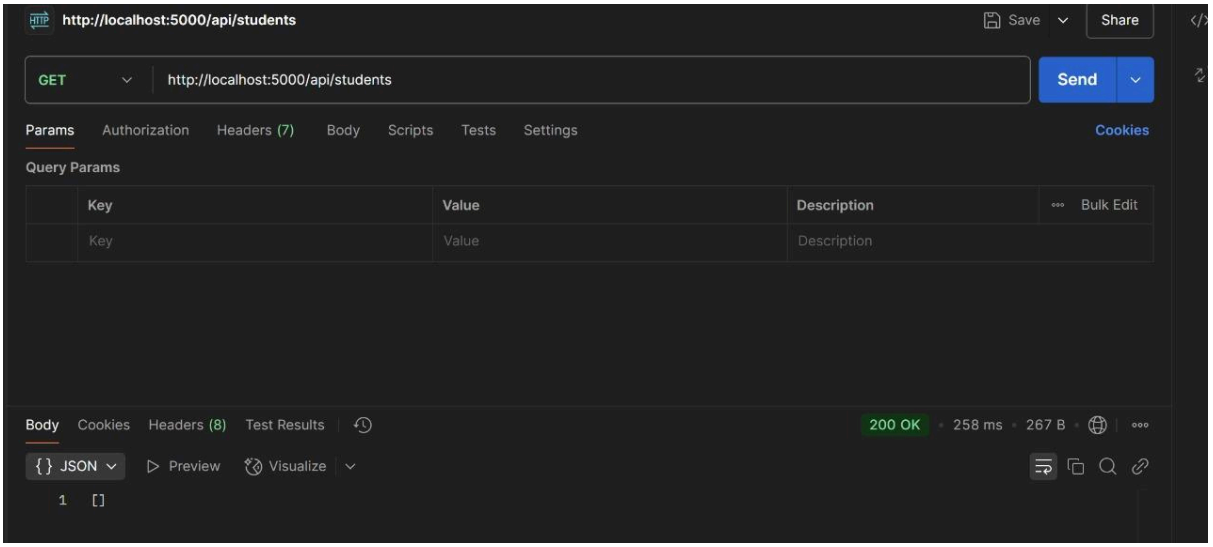
JS server.js > useUnifiedTopology

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const studentRoutes = require('./routes/studentRoutes');

const app = express();
app.use(cors());
app.use(express.json());

mongoose.connect('mongodb+srv://riachaudhari68:centralorchid17@cluster0.lzxhhat.mongodb.net/?retryWrites=true&w=majority&appNam
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => console.log('MongoDB connected'));

app.use('/api/students', studentRoutes);

app.listen(5000, () => {
  console.log('Server running on http://localhost:5000');
});
```

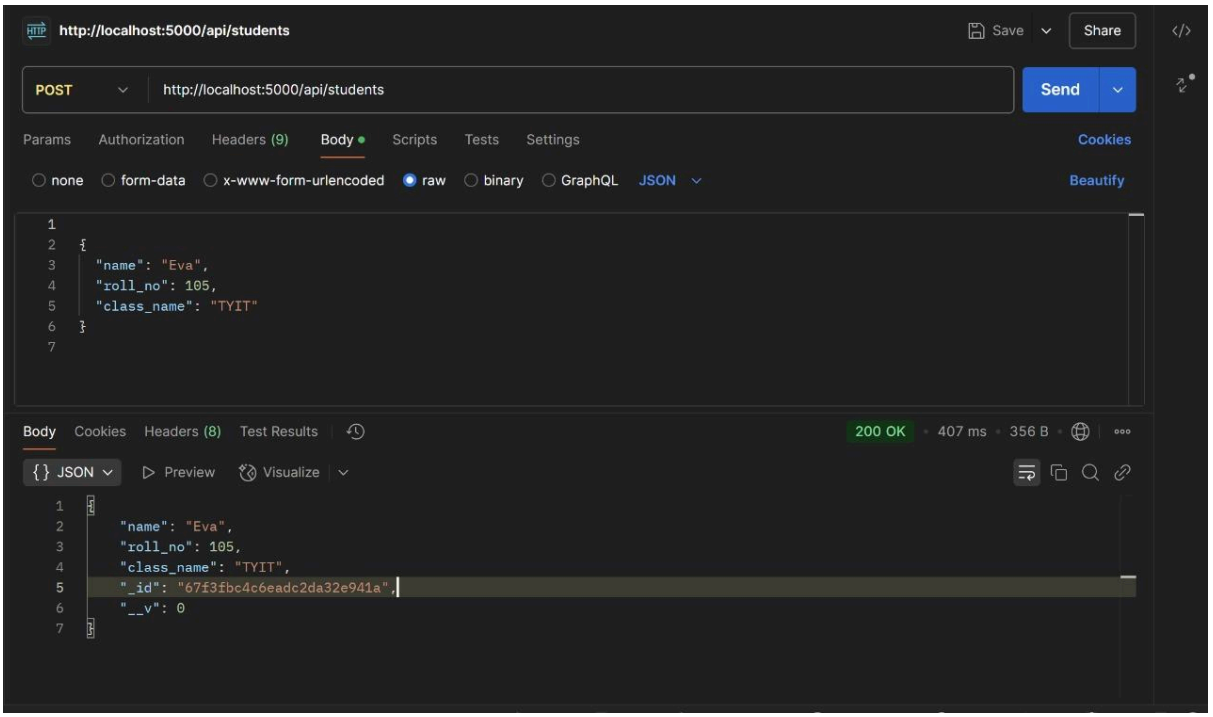
PROBLEMS DEBUG CONSOLE TERMINAL PORTS OUTPUT bash

```
(Use `node --trace-warnings ...` to show where the warning was created)
(node:22392) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver v
d will be removed in the next major version
Server running on http://localhost:5000
MongoDB connected

Ria Chaudhari@Ria MINGW64 ~/webx/student-api (master)
$ node server.js
(node:20972) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version
 be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:20972) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver v
d will be removed in the next major version
Server running on http://localhost:5000
MongoDB connected
```

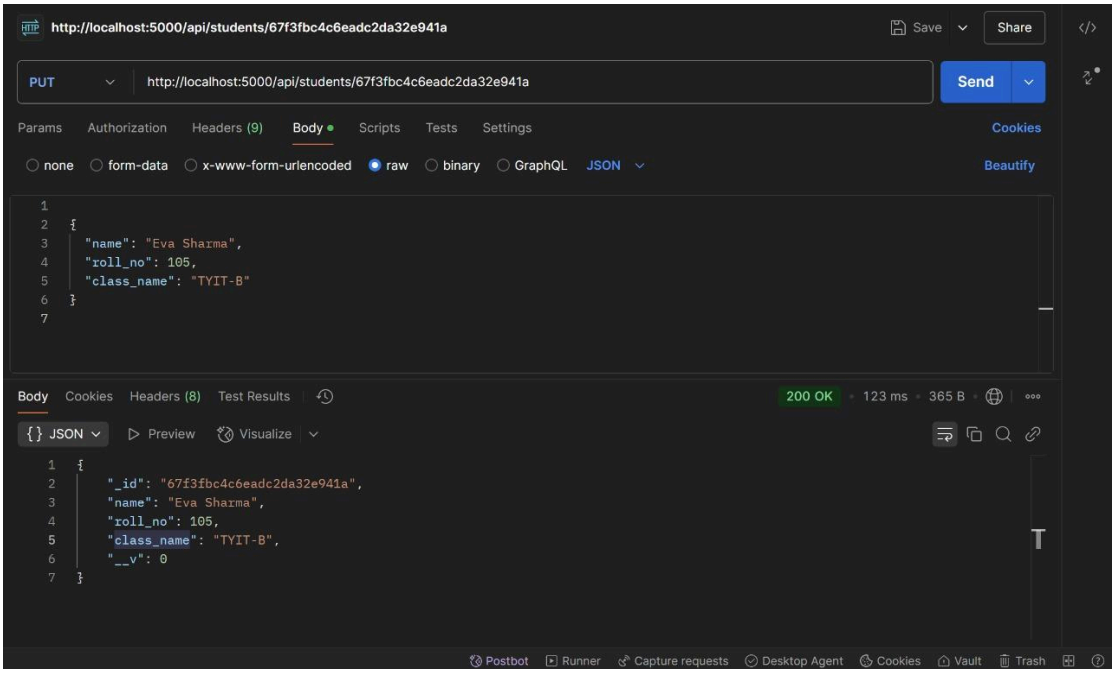
a) Retrieve details



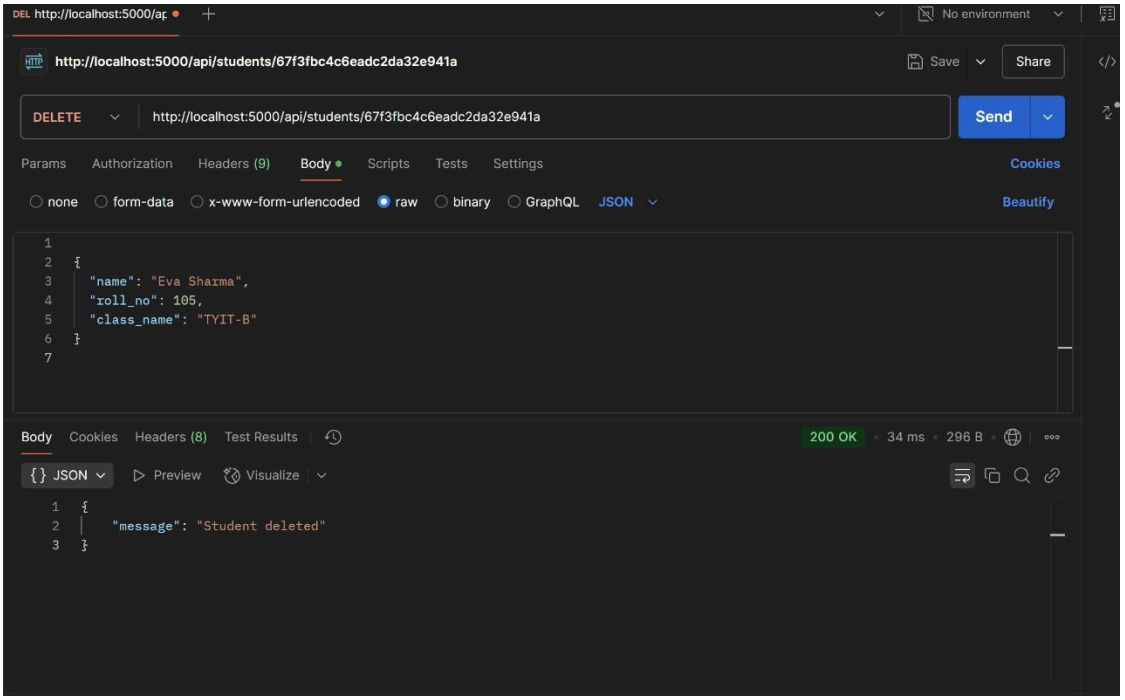
b) Add a new student to the database



c) Update details of an existing student by ID.



d) 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.