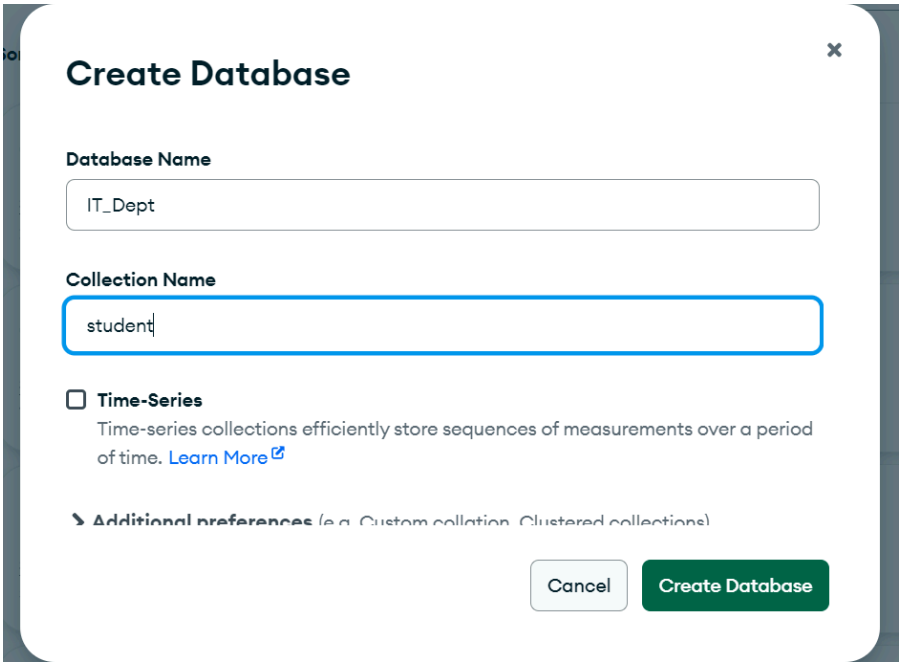
Experiment – 7: MongoDB

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

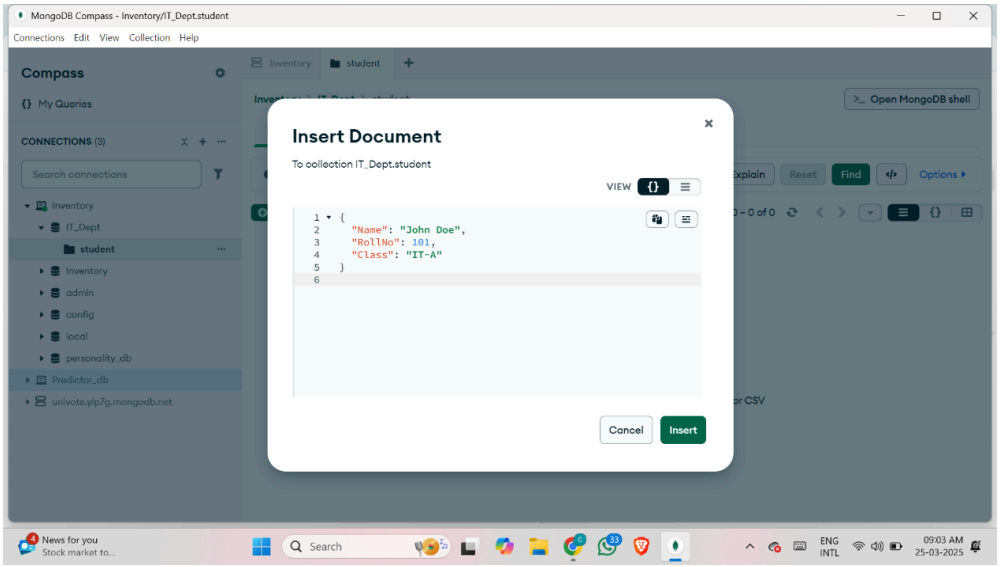
   B) Create a set of RESTful endpoints using Node.js, Express, and Mongoose for handling student data operations.
   The endpoints should support:
   ● Retrieve a list of all students.
   ● Retrieve details of an individual student by ID.
   ● Add a new student to the database.
   ● Update details of an 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:**

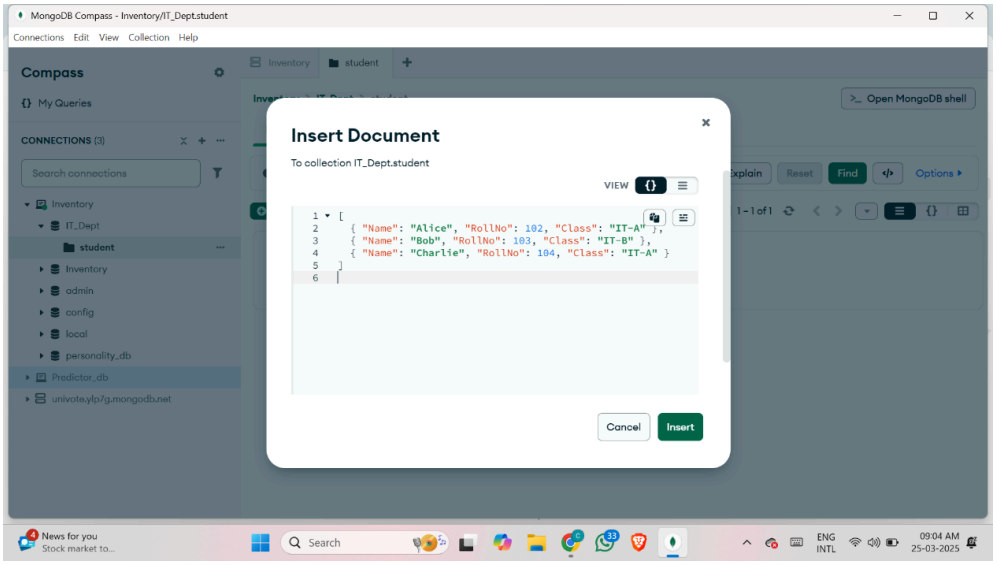
Part A: MongoDB CRUD Operations in Compass. 1) Create Database and Collection



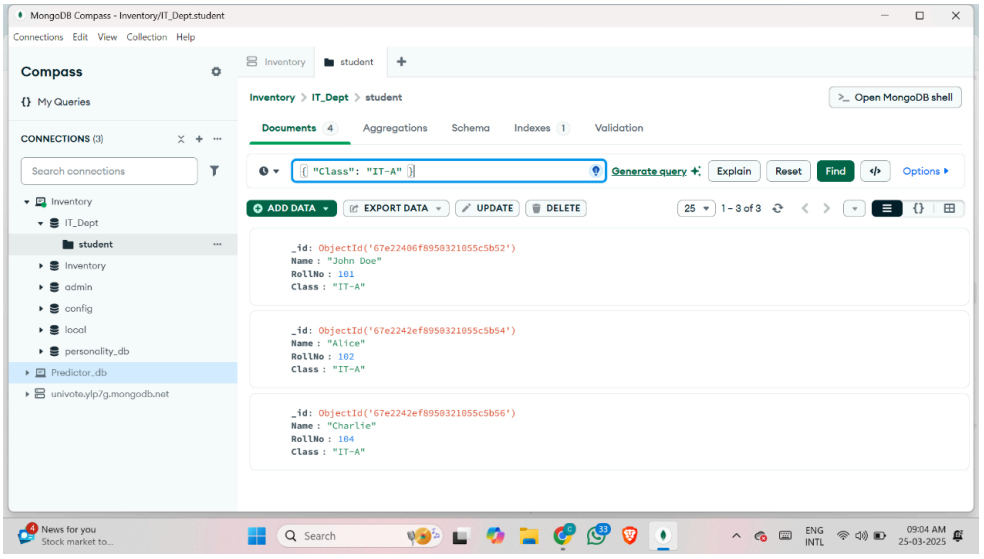
2) Insert One Student Record



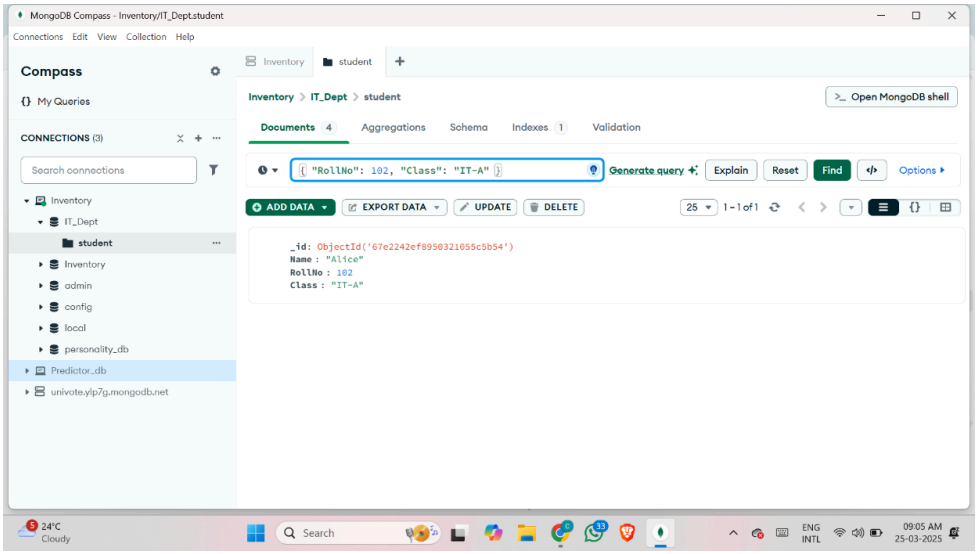
3) Insert Multiple Students at Once



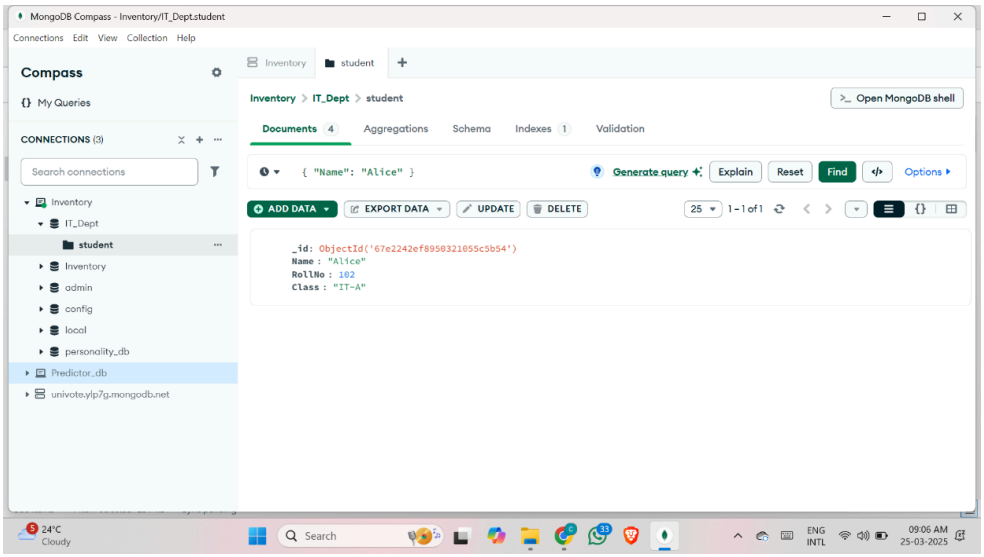
4) Display Students for a Particular Class

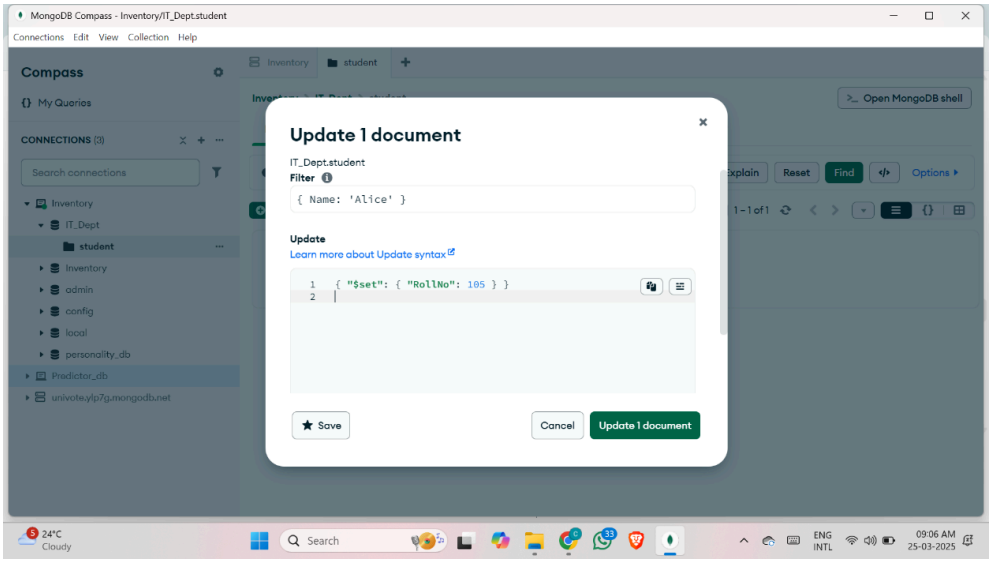


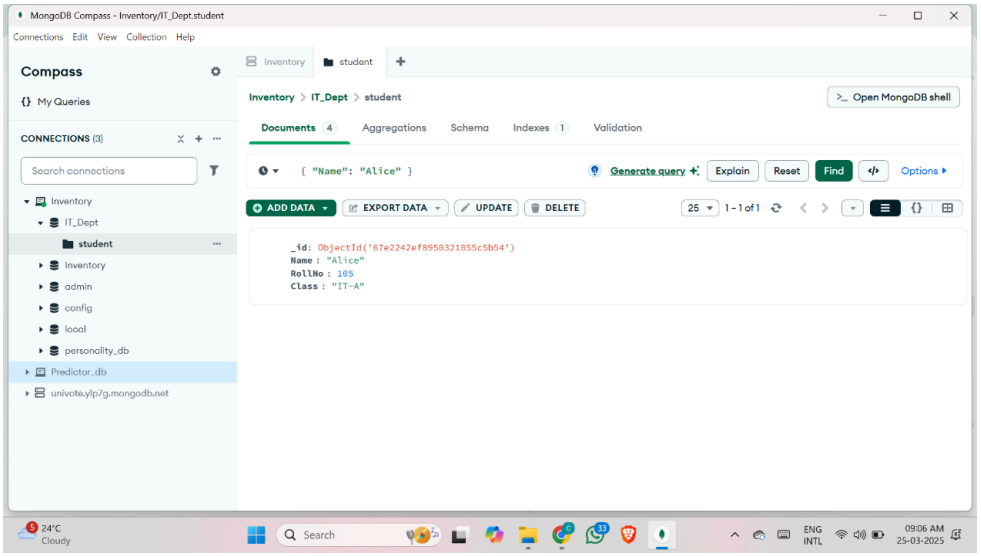
5) Display a Student With a Specific Roll No in a Class



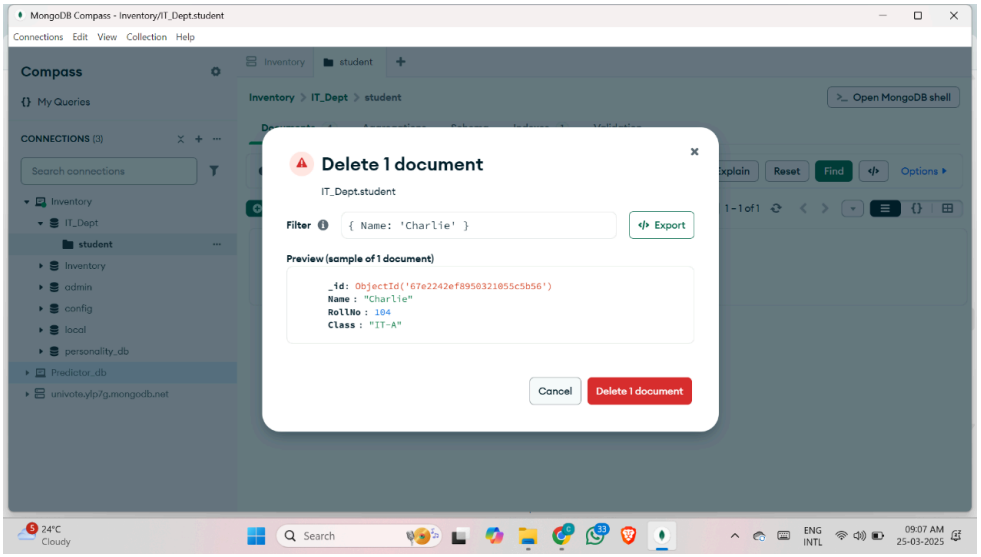
6) Change the Roll No of a Student

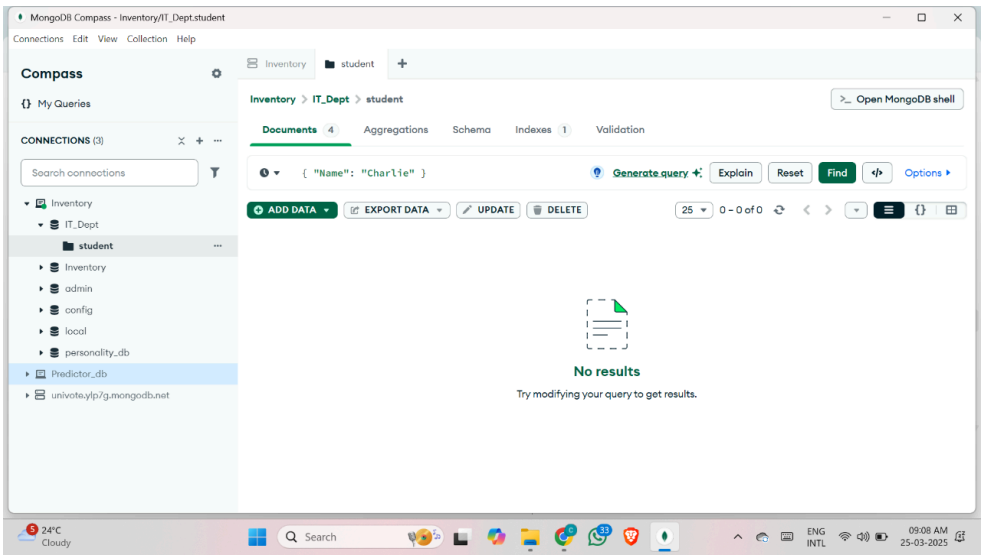


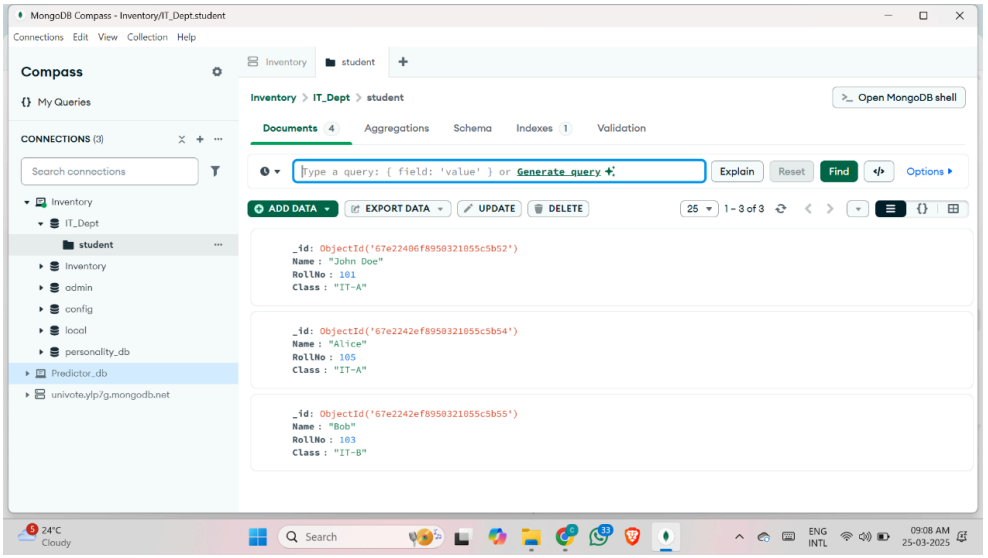




7) Delete a Particular Student Entry






MongoDB Compass - Inventory/IT_Dept.student
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (3)
Search connections
Inventory
IT_Dept
student
Inventory
admin
config
local
personality_db
Predictor_db
univote.ylp7g.mongodb.net
Inventory
student
Inventory > IT_Dept > student
Open MongoDB shell
Documents 4
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 3 of 3
_id: ObjectId('67e22406f8958321055c5b52')
Name : "John Doe"
RollNo : 101
Class : "IT-A"
_id: ObjectId('67e2242ef8958321055c5b54')
Name : "Alice"
RollNo : 105
Class : "IT-A"
_id: ObjectId('67e2242ef8958321055c5b55')
Name : "Bob"
RollNo : 103
Class : "IT-B"
24°C
Cloudy
Search
ENG
INTL
09:08 AM
25-03-2025

Part B: Creating RESTful API with Node.js, Express, and Mongoose

server.js:

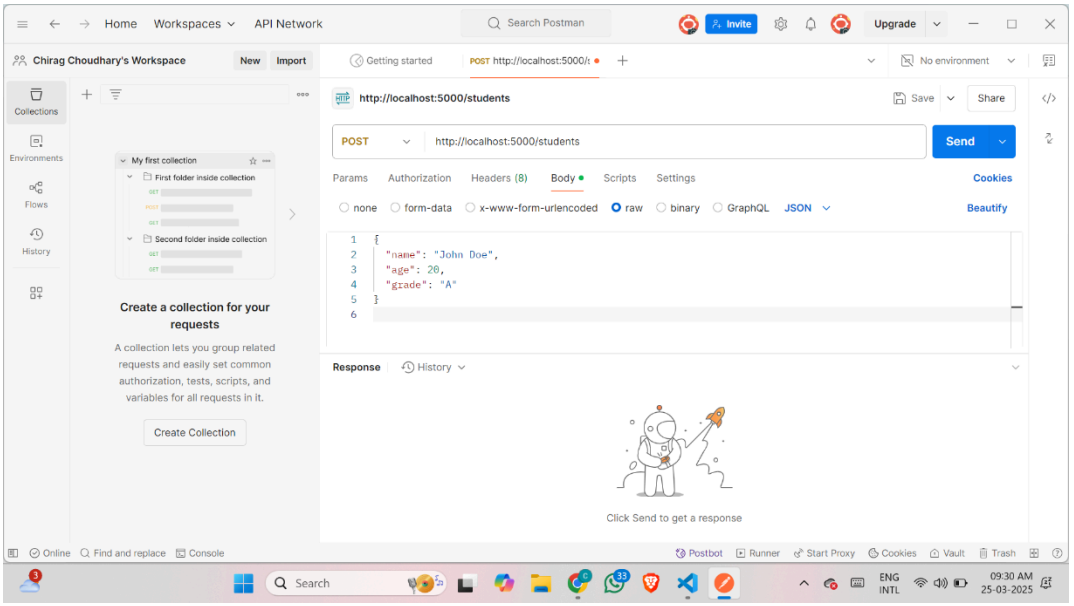
```
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');
const app = express();
app.use(bodyParser.json());
app.use(cors());
// MongoDB Connection
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log('MongoDB Connected'))
  .catch(err => console.error('MongoDB Connection Failed:', err));
// Define Student Schema
const studentSchema = new mongoose.Schema({
  name: String,
  age: Number,
  grade: String
});

const Student = mongoose.model('Student', studentSchema);
// Routes
app.get('/students', async (req, res) => {
  const students = await Student.find();
  res.json(students);
});
app.get('/students/:id', async (req, res) => {
  try {
    const student = await Student.findById(req.params.id);
    res.json(student);
  } catch (err) {
    res.status(404).json({ error: 'Student not found' });
  }
});
app.post('/students', async (req, res) => {
  const newStudent = new Student(req.body);
  await newStudent.save();
  res.json(newStudent);
});
app.put('/students/:id', async (req, res) => {
  try {
    const updatedStudent = await Student.findByIdAndUpdate(req.params.id, req.body, { new: true });
    res.json(updatedStudent);
  } catch (err) {
    res.status(404).json({ error: 'Student not found' });
  }
});
app.delete('/students/:id', async (req, res) => {
  try {
    await Student.findByIdAndDelete(req.params.id);
```

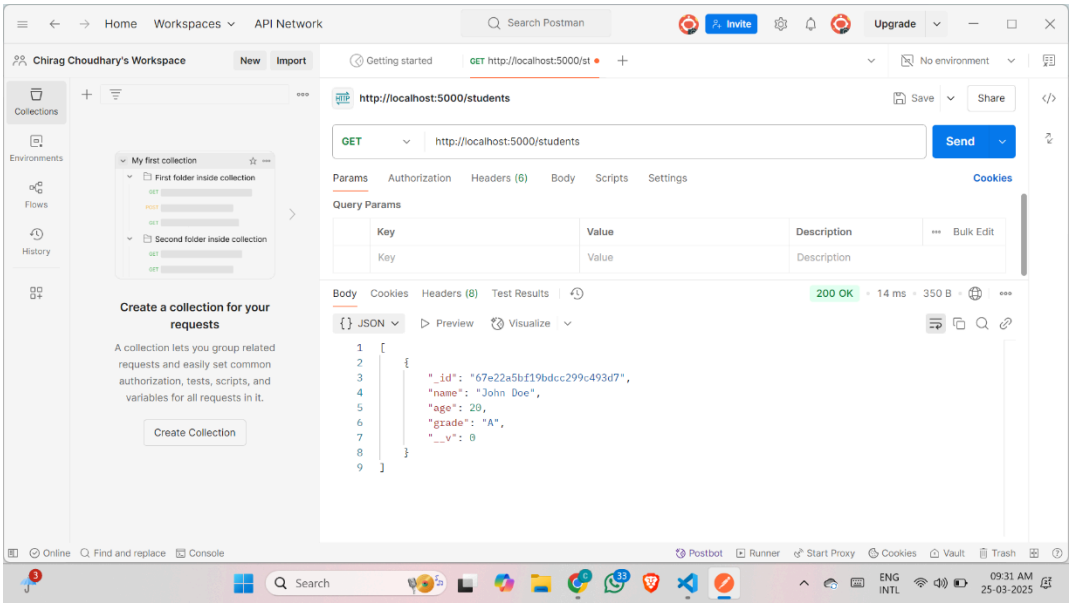
```
    res.json({ message: 'Student deleted' });
  } catch (err) {
    res.status(404).json({ error: 'Student not found' });
  }
});
// Server Start
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

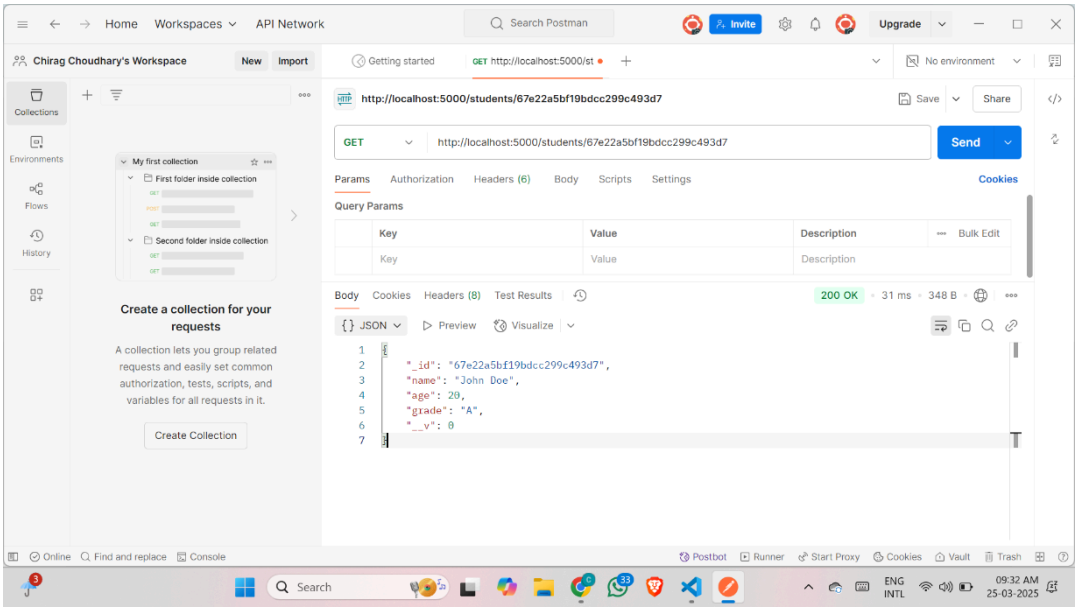
.env:

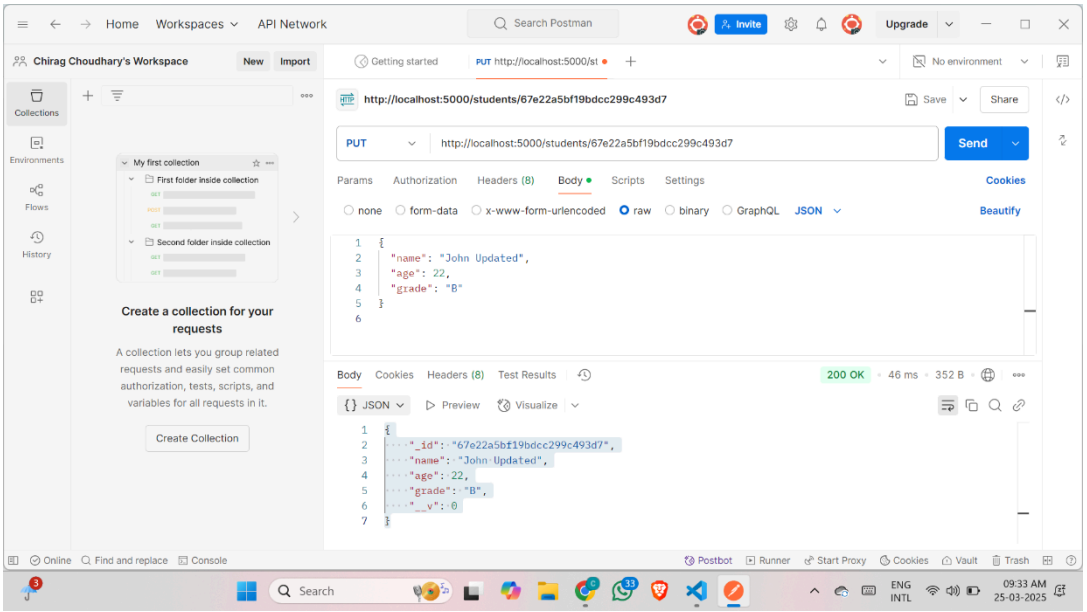
```
MONGO_URI=mongodb://localhost:27017/IT_Dept
PORT=5000
```

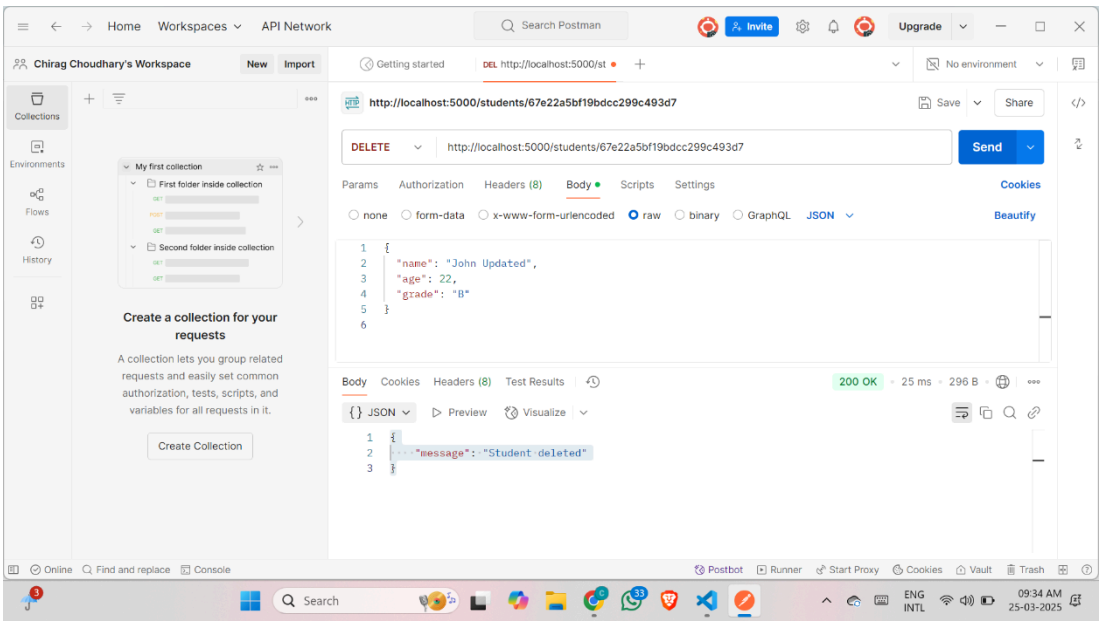
Postman API Testing :






Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Chirag Choudhary's Workspace
New
Import
Getting started
GET http://localhost:5000/st
No environment
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Save
Share
GET
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (8)
Test Results
200 OK
31 ms
348 B
JSON
Preview
Visualize
{
"_id": "67e22a5bf19bdcc299c493d7",
"name": "John Doe",
"age": 20,
"grade": "A",
"__v": 0
}
Collections
Environments
Flows
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
Search
ENG
INTL
09:32 AM
25-03-2025


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Chirag Choudhary's Workspace
New
Import
Getting started
PUT http://localhost:5000/st
No environment
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Save
Share
PUT
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "John Updated",
"age": 22,
"grade": "B"
}
Body
Cookies
Headers (8)
Test Results
200 OK
46 ms
352 B
JSON
Preview
Visualize
{
"_id": "67e22a5bf19bdcc299c493d7",
"name": "John Updated",
"age": 22,
"grade": "B",
"__v": 0
}
Collections
Environments
Flows
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
Search
ENG
INTL
09:33 AM
25-03-2025


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Chirag Choudhary's Workspace
New
Import
Getting started
DEL http://localhost:5000/st
No environment
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Save
Share
DELETE
http://localhost:5000/students/67e22a5bf19bdcc299c493d7
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "John Updated",
"age": 22,
"grade": "B"
}
Body
Cookies
Headers (8)
Test Results
200 OK
25 ms
296 B
JSON
Preview
Visualize
{
"message": "Student deleted"
}
Collections
Environments
Flows
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
Search
ENG
INTL
09:34 AM
25-03-2025