Experiment 7: MongoDB CRUD Operations and RESTful API Development

Name of Student	Diksha Utekar
Class Roll No.	D15A/63
DOP	
DOS	
Sign and Grade	

Aim:

To study CRUD operations in MongoDB and implement RESTful APIs using Node.js, Express, and Mongoose.

Part A: MongoDB CRUD Operations

Create a database to store student details (Name, Roll No, Class Name) and perform the following operations:

1. Insert a single student record

2. Insert multiple student records at once

3. Display students of a particular class

4. Retrieve a student by Roll No

5. Update the Roll No of a student

6. Delete a student record

Implementation:
- Use MongoDB Compass or CLI to create a database and perform CRUD operations.
- Sample queries for inserting, updating, and deleting records.

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

API Endpoints:

- GET /students - Retrieve all students

- GET /students/:id - Retrieve a student by ID

- POST /students - Add a new student

- PUT /students/:id - Update an existing student

- DELETE /students/:id - Remove a student from the database

Server Implementation (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());

mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log('MongoDB Connected'))
  .catch(err => console.error('MongoDB Connection Failed:', err));

const studentSchema = new mongoose.Schema({ name: String, age: Number, grade: String });
const Student = mongoose.model('Student', studentSchema);

app.get('/students', async (req, res) => { res.json(await Student.find()); });
app.get('/students/:id', async (req, res) => { res.json(await Student.findById(req.params.id)); });
app.post('/students', async (req, res) => { res.json(await new Student(req.body).save()); });
app.put('/students/:id', async (req, res) => { res.json(await Student.findByIdAndUpdate(req.params.id, req.body, { new: true })); });
app.delete('/students/:id', async (req, res) => { await Student.findByIdAndDelete(req.params.id); res.json({ message: 'Deleted' }); });

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

Conclusion:

Successfully implemented CRUD operations using MongoDB.Developed a RESTful API using Node.js, Express, and Mongoose.Demonstrated database connectivity and API routing for student records management.