**CODE : -**

**Student / Model.js**

const mongoose = require('mongoose');

const studentSchema = new mongoose.Schema({

name: {

type: String,

required: true

},

age: Number,

department: String

});

module.exports = mongoose.model('Student', studentSchema);

**Student / routes.js**

const express = require('express');

const router = express.Router();

const Student = require('../models/students');

// CREATE

router.post('/', async (req, res) => {

try {

const student = new Student(req.body);

await student.save();

res.status(201).json(student);

} catch (error) {

res.status(400).json({ error: error.message });

}

});

// READ ALL

router.get('/', async (req, res) => {

try {

const students = await Student.find();

res.json(students);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// READ ONE

router.get('/:id', async (req, res) => {

try {

const student = await Student.findById(req.params.id);

if (!student) return res.status(404).json({ message: 'Student not found' });

res.json(student);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// UPDATE

router.put('/:id', async (req, res) => {

try {

const student = await Student.findByIdAndUpdate(req.params.id, req.body, { new: true });

res.json(student);

} catch (error) {

res.status(400).json({ error: error.message });

}

});

// DELETE

router.delete('/:id', async (req, res) => {

try {

await Student.findByIdAndDelete(req.params.id);

res.json({ message: 'Student deleted' });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

module.exports = router;

**index.js**

const express = require('express');

const mongoose = require('mongoose');

const app = express();

app.use(express.json());

mongoose.connect('mongodb://127.0.0.1:27017/studentdb', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

const studentSchema = new mongoose.Schema({

name: String,

age: Number,

department: String,

});

const Student = mongoose.model('Student', studentSchema);

app.post('/students', async (req, res) => {

const student = new Student(req.body);

await student.save();

res.send(student);

});

app.get('/students', async (req, res) => {

const students = await Student.find();

res.send(students);

});

app.listen(3001, () => {

console.log('Server is running on http://localhost:3001');

});

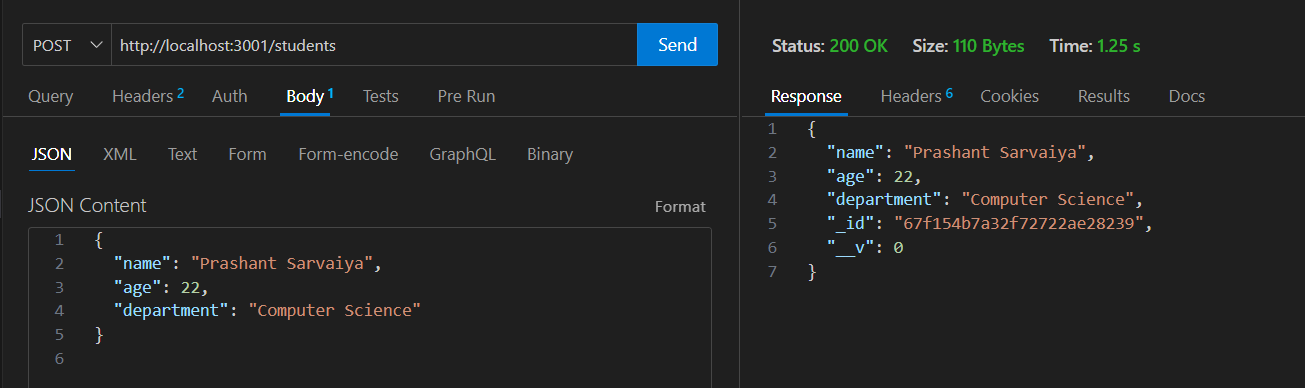
**.env**

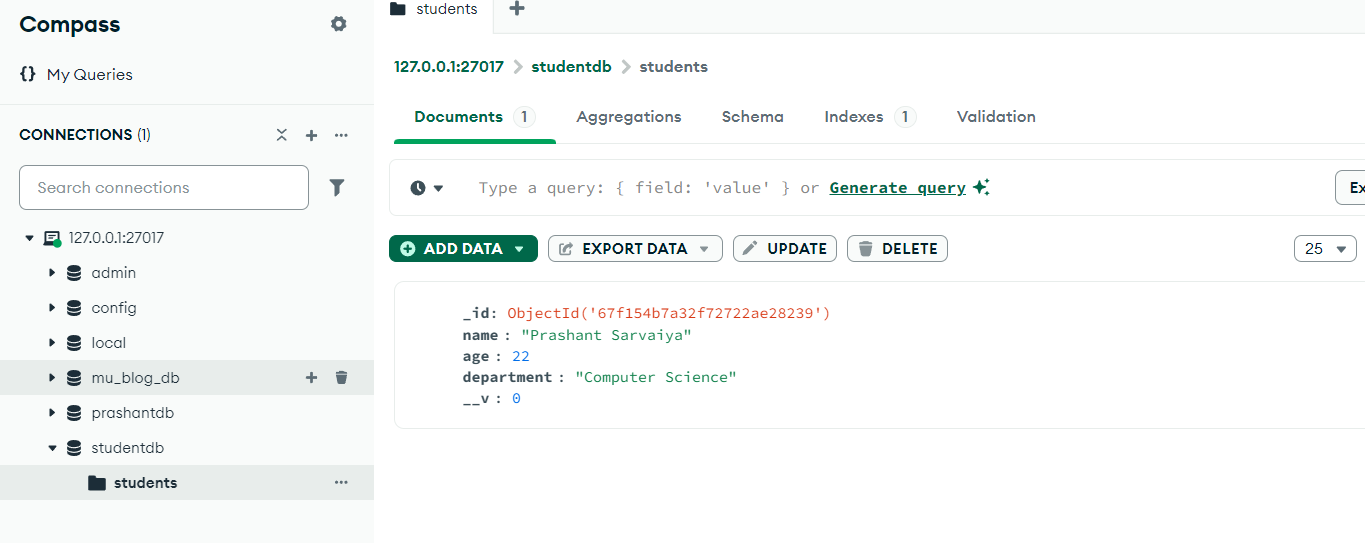
MONGO\_URI=mongodb://localhost:27017/studentdb

PORT=3001

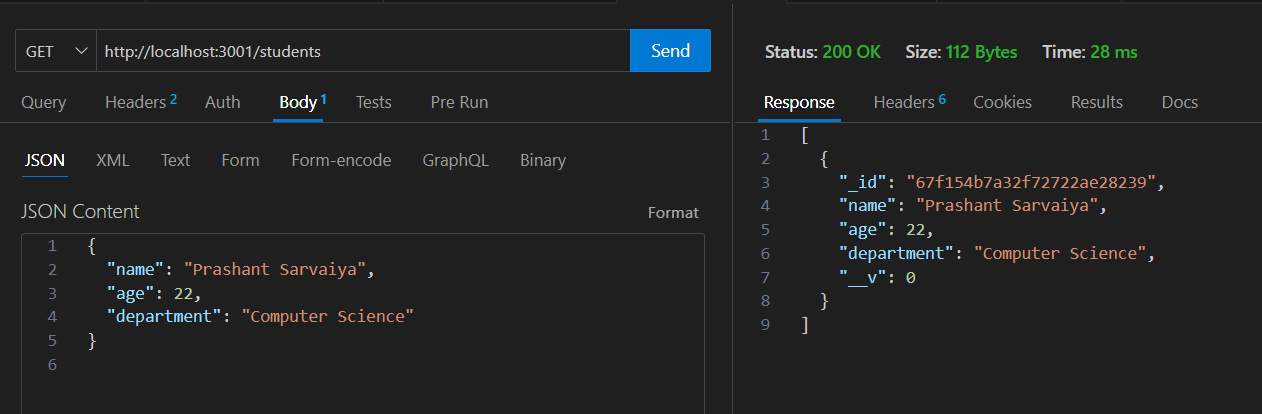
**OUTPUT:**

**POST Method**

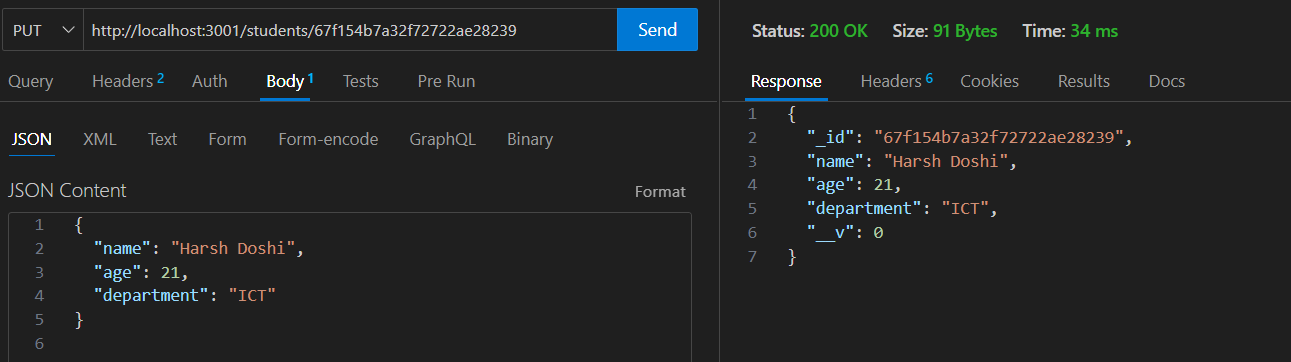


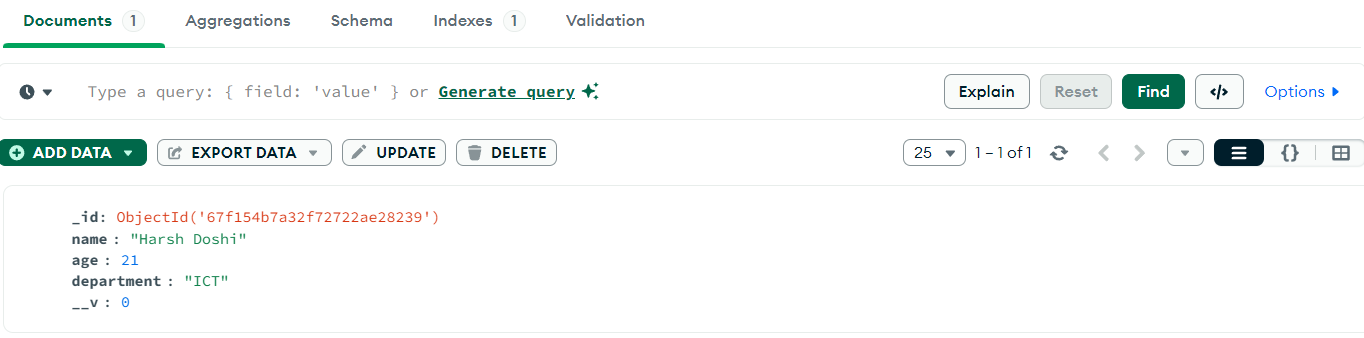


**GET Method**



**PUT Method:**





**DELETE Method**

