**CODE :-**

**index.js**

const express = require("express");

const app = express();

const port = 3001;

const cors = require("cors");

app.use(

cors({

origin: "http://localhost:5173",

})

);

//middleware

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

const dbConfig = require("./src/config/db.config");

const mongoose = require("mongoose");

mongoose.Promise = global.Promise;

mongoose

.connect(dbConfig.url, {

useNewUrlParser: true,

useUnifiedTopology: true,

})

.then(() => {

console.log("connection done with database");

})

.catch((err) => {

console.log("error in db connection ", err);

process.exit();

});

const blogRouter = require("./src/blog/blog.routes");

app.use("/api/blog", blogRouter);

app.get("/", (req, res) => res.send("Hello World!"));

app.listen(port, () => console.log(`Example app listening on port ${port}!`));

**blog.controller.js**

const blogService = require("./blog.service");

exports.create = async (req, res) => {

const { title, description, author } = req.body;

if (!title || !description || !author) {

return res.status(400).json({ message: "Fields are empty" });

}

try {

const blog = await blogService.createBlog(title, description, author);

res.status(201).json({ message: "Blog created successfully", blog });

} catch (err) {

console.error("Error creating blog:", err);

res.status(500).json({ error: "Internal server error" });

}

};

exports.findAll = async (req, res) => {

try {

const blogs = await blogService.findAllBlogs();

res.send({ message: "Blogs data", blogs });

} catch (err) {

console.error("Error fetching blogs:", err);

res.status(500).json({ error: "Internal server error" });

}

};

exports.findAllByTitle = async (req, res) => {

const title = req.query.title;

try {

const blogs = await blogService.findBlogsByTitle(title);

res.send({ message: "Blogs data", blogs });

} catch (err) {

console.error("Error fetching blogs by title:", err);

res.status(500).json({ error: "Internal server error" });

}

};

exports.findOne = async (req, res) => {

const id = req.params.id;

try {

const blog = await blogService.findBlogById(id);

if (!blog) {

return res.status(400).json({ message: "No blog found with id: " + id });

}

res.json({ message: "Data found", blog });

} catch (err) {

console.error("Error fetching blog by id:", err);

res.status(500).json({ error: "Internal server error" });

}

};

exports.update = async (req, res) => {

const id = req.params.id;

const updateData = req.body;

try {

const result = await blogService.updateBlog(id, updateData);

if (!result) {

return res.status(400).json({ message: "No blog found with id: " + id });

}

res.json({ message: "Blog updated successfully" });

} catch (err) {

console.error("Error updating blog:", err);

res.status(500).json({ error: "Internal server error" });

}

};

exports.delete = async (req, res) => {

const id = req.params.id;

try {

const result = await blogService.deleteBlog(id);

if (!result) {

return res.status(400).json({ message: "No blog found with id: " + id });

}

res.json({ message: "Blog deleted successfully" });

} catch (err) {

console.error("Error deleting blog:", err);

res.status(500).json({ error: "Internal server error" });

}

};

**blog.model.js**

const mongoose = require("mongoose");

var schema = new mongoose.Schema(

{

title: {

type: String,

required: true,

},

description: {

type: String,

required: true,

},

author: {

type: String,

required: true,

},

},

{

timestamps: true,

}

);

var blog = new mongoose.model("mublog", schema);

module.exports = blog;

**blog.routes.js**

const blog = require("./blog.controller");

const express = require("express");

const router = express.Router();

router.post("/", blog.create);

router.get("/", blog.findAll);

router.get("/title/", blog.findAllByTitle);

router.get("/:id", blog.findOne);

router.put("/:id", blog.update);

router.delete("/:id", blog.delete);

module.exports = router;

**blog.service.js**

const BlogModel = require("./blog.model");

exports.createBlog = async (title, description, author) => {

const blog = new BlogModel({

title,

description,

author,

});

return await blog.save();

};

exports.findAllBlogs = async () => {

return await BlogModel.find();

};

exports.findBlogsByTitle = async (title) => {

const condition = title

? { title: { $regex: new RegExp(title), $options: "i" } }

: {};

return await BlogModel.find(condition);

};

exports.findBlogById = async (id) => {

return await BlogModel.findById(id);

};

exports.updateBlog = async (id, updateData) => {

return await BlogModel.findByIdAndUpdate(id, updateData, {

useFindAndModify: false,

});

};

exports.deleteBlog = async (id) => {

return await BlogModel.findByIdAndDelete(id);

};

**db.config.js**

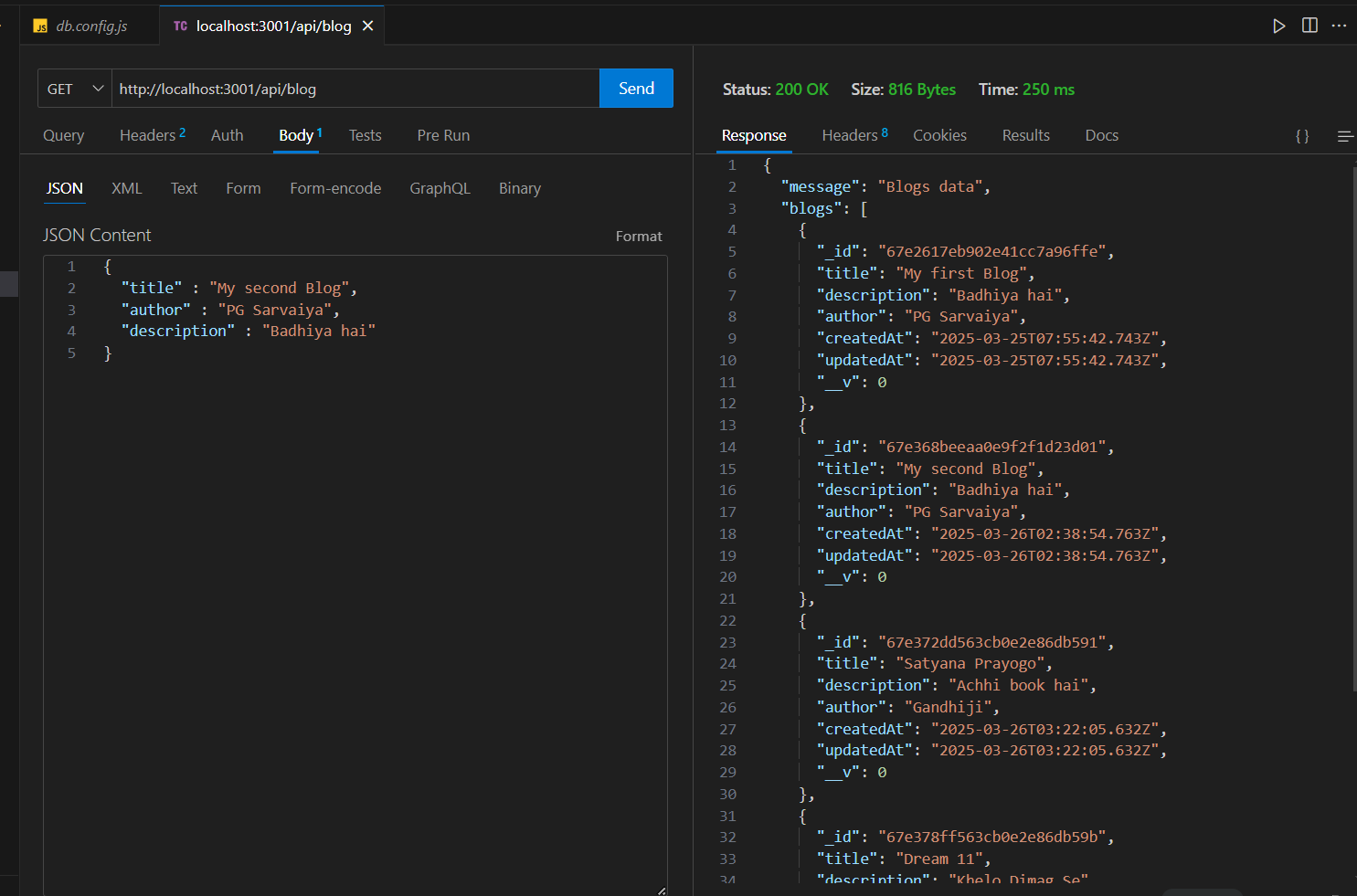
module.exports = {

url: "mongodb://127.0.0.1:27017/mu\_blog\_db",

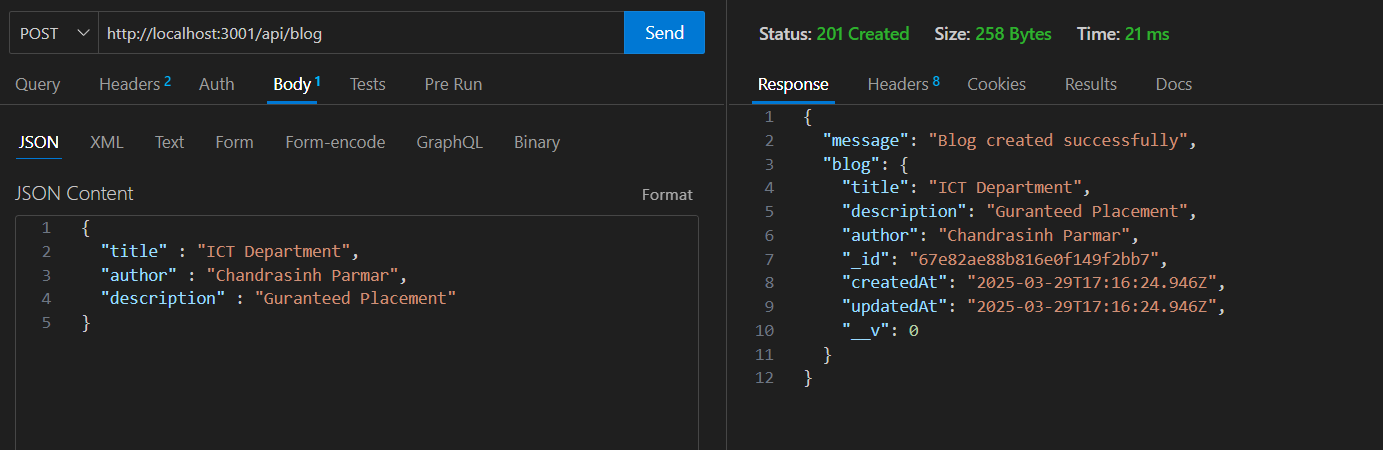
};

**OUTPUT :-**

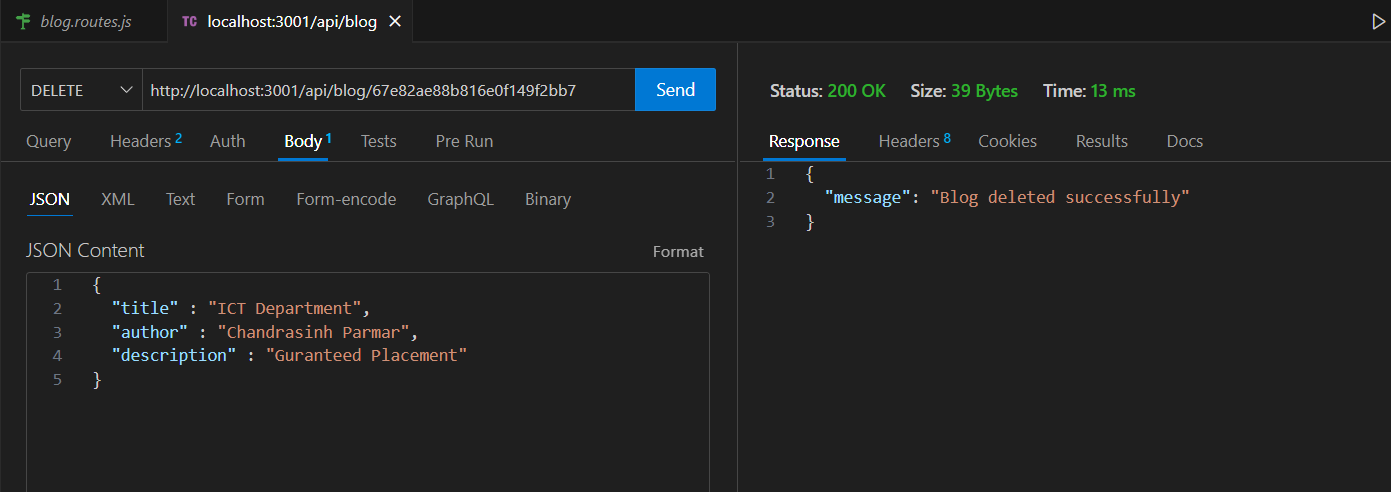
**Get**



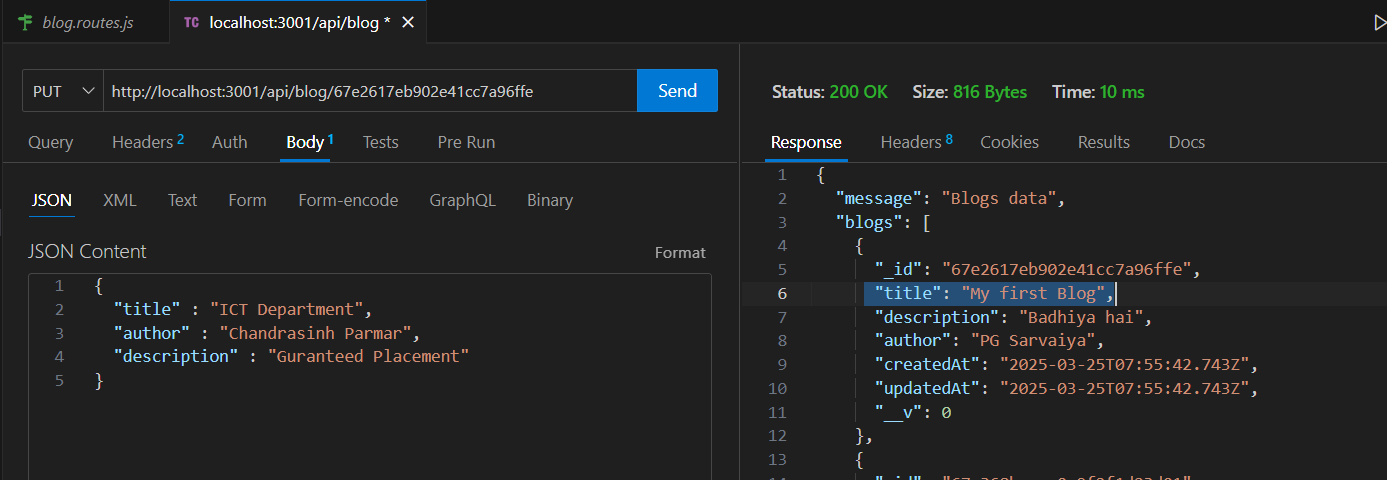
**POST**



**DELETE :**



**PUT:**



**MONGODB :**

