**CODE :**

**Index.js**

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const userRoutes = require('./routes/userRoutes');

const app = express();

const PORT = 5000;

// Middleware

app.use(bodyParser.json());

// MongoDB Connection

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

useNewUrlParser: true,

useUnifiedTopology: true,

})

.then(() => console.log('MongoDB Connected!'))

.catch((err) => console.log(err));

// Routes

app.use('/api/users', userRoutes);

// Start Server

app.listen(PORT, () => {

console.log(`Server is running on port ${PORT}`);

});

**Model**

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

name: {

type: String,

required: true,

},

email: {

type: String,

required: true,

unique: true,

},

age: Number,

});

module.exports = mongoose.model('User', userSchema);

**Route**

const express = require('express');

const router = express.Router();

const User = require('../model/user');

// CREATE

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

try {

const user = new User(req.body);

const savedUser = await user.save();

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

} catch (err) {

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

}

});

// READ All

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

try {

const users = await User.find();

res.json(users);

} catch (err) {

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

}

});

// READ One

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

try {

const user = await User.findById(req.params.id);

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

res.json(user);

} catch (err) {

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

}

});

// UPDATE

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

try {

const updatedUser = await User.findByIdAndUpdate(

req.params.id,

req.body,

{ new: true } // return updated doc

);

res.json(updatedUser);

} catch (err) {

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

}

});

// DELETE

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

try {

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

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

} catch (err) {

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

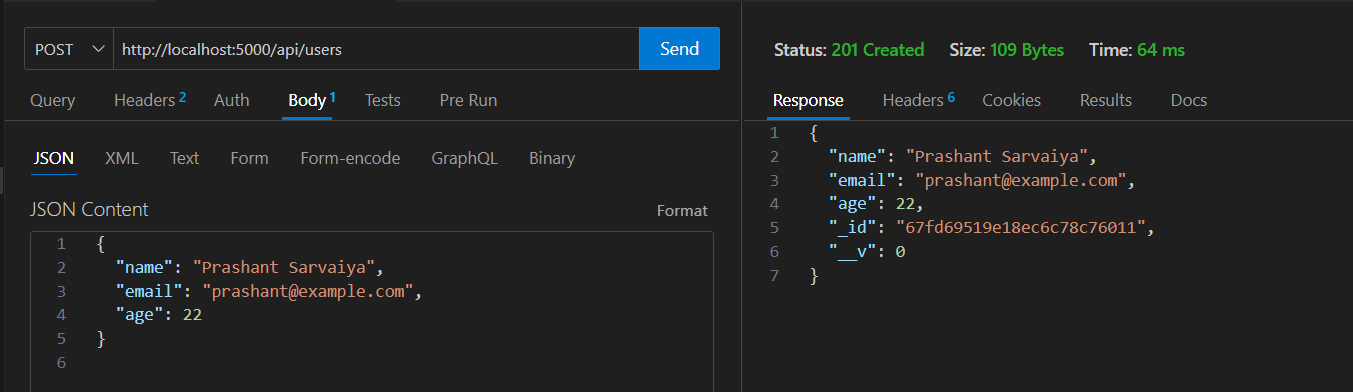
}

});

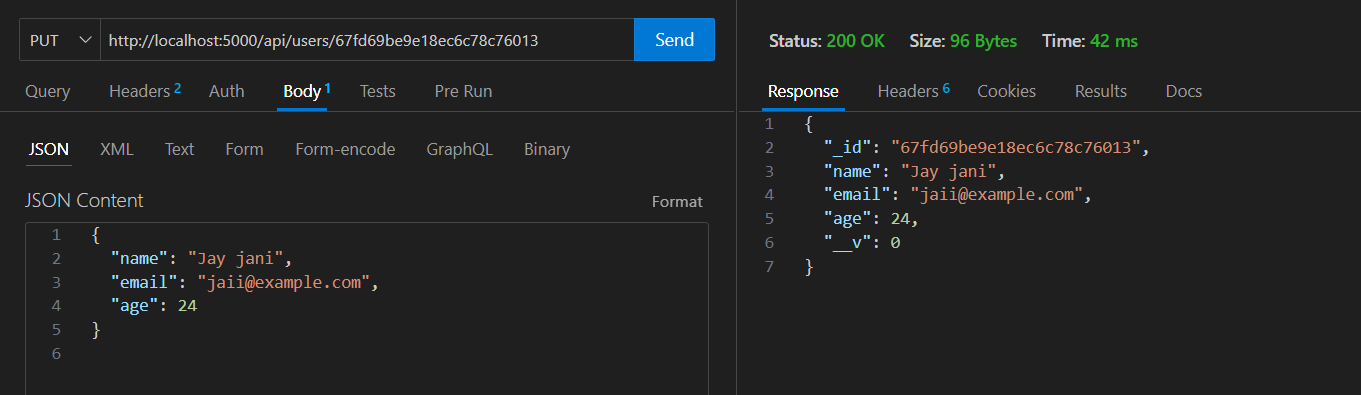
module.exports = router;

**OUTPUT**

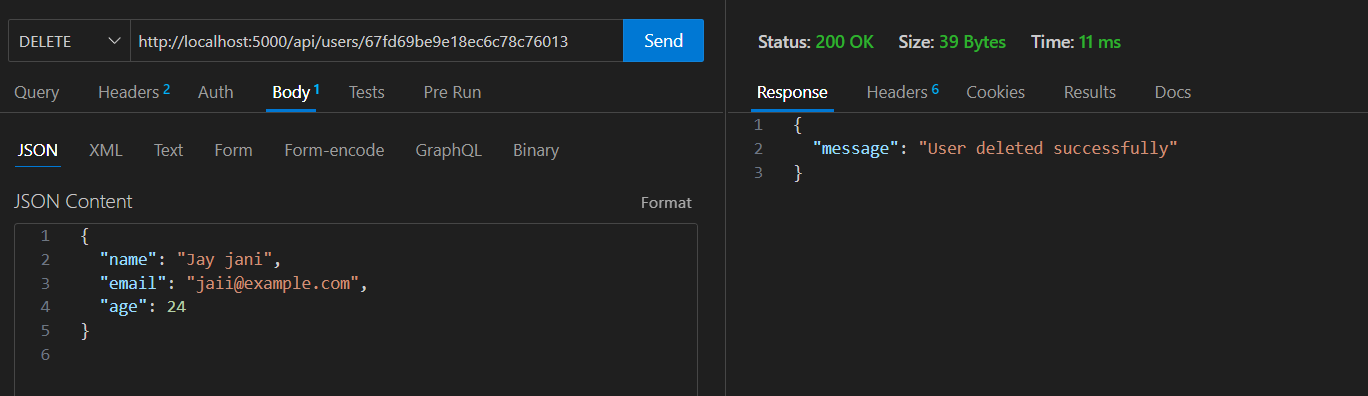
**POST**



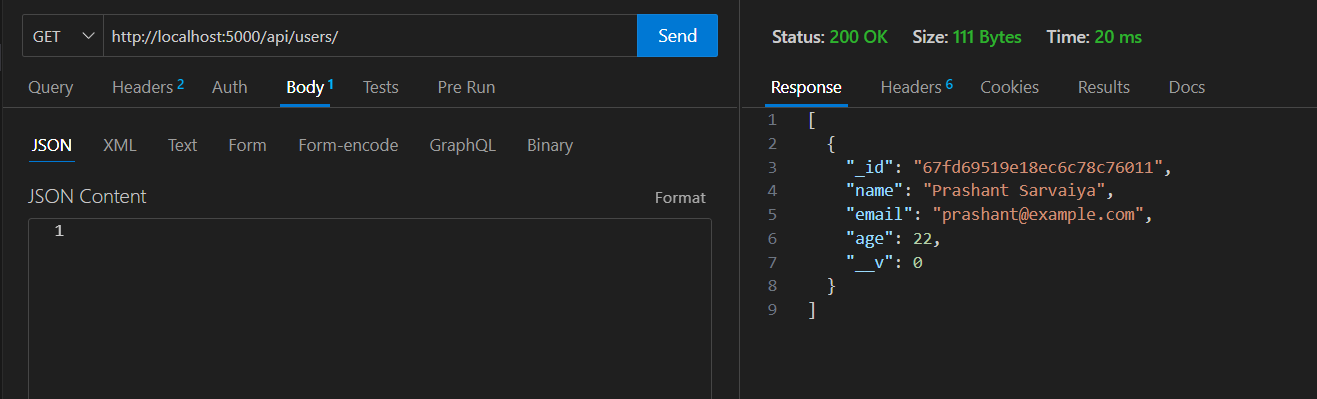
**UPDATE**



**DELETE**



**GET**



**DB**

