FULL STACK WITH MERN (JAVA) ASSIGNMENT-3

Sanagala Sai Sreeja
Vignan's Nirula Institute of
Technology and Science
For Women (VNITSW)
4th B-Tceh (CSE)
20NN1A05A2

ASSIGNMENT-3: Create a RESTful API using express.js and create a database and index in MongoDB

1. INDEX.JS:

```
// index.js

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

const app = express();
const PORT = process.env.PORT || 4000;

// Middleware
app.use(bodyParser.json());

// Routes
const todoRoutes = require('./routes/todoRoutes');
app.use('/api/todos', todoRoutes);

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/todoDB', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => {
    console.log('Connected to MongoDB');
    app.listen(PORT, () => {
      console.log(`Server is running on port ${PORT}`);
    });
  })
  .catch(err => console.error('Error connecting to MongoDB:', err));
```

2. ROUTES (todoRoutes.js):

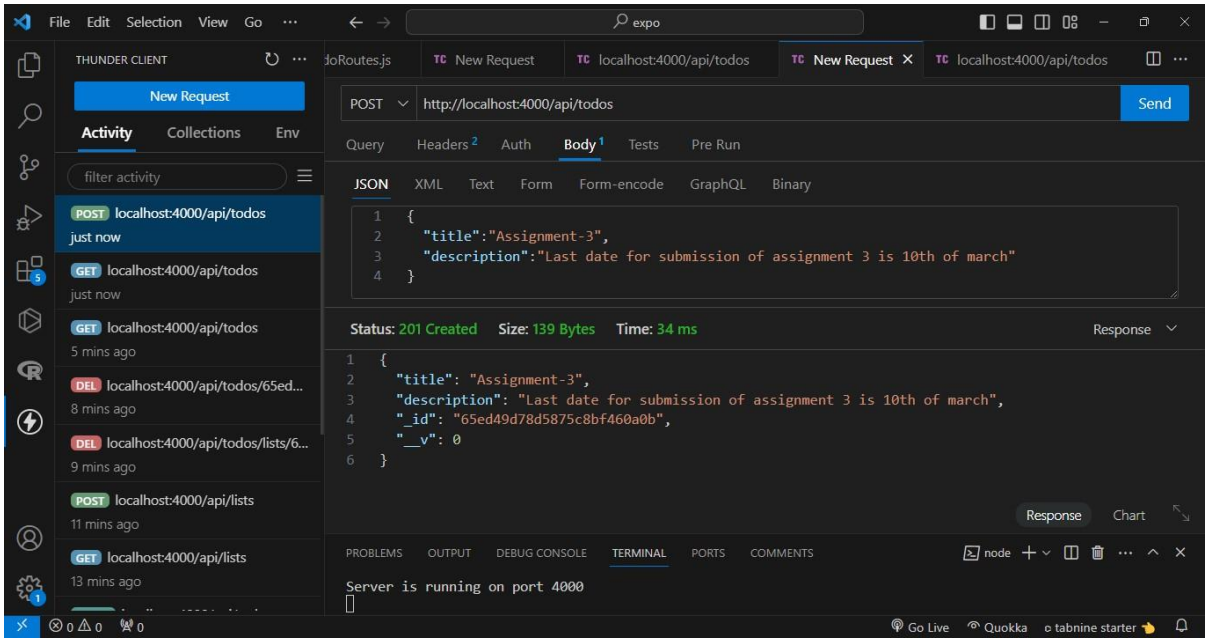
```
// routes/todoRoutes.js
const express = require('express');
const router = express.Router();
const Todo = require('../models/todoModel');

// Create a new todo
router.post('/', async (req, res) => {
  try {
    const todo = await Todo.create(req.body);
    res.status(201).json(todo);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

// Read all todos
router.get('/', async (req, res) => {
  try {
    const todos = await Todo.find();
    res.json(todos);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

// Update a todo
router.patch('/:id', async (req, res) => {
  try {
    const todo = await Todo.findByIdAndUpdate(req.params.id, req.body, { new:
true });
    res.json(todo);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});
// Delete a todo
router.delete('/:id', async (req, res) => {
  try {
    await Todo.findByIdAndDelete(req.params.id);
    res.json({ message: 'Todo deleted' });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});


module.exports = router;
```

3. MODELS (todoModel.js):

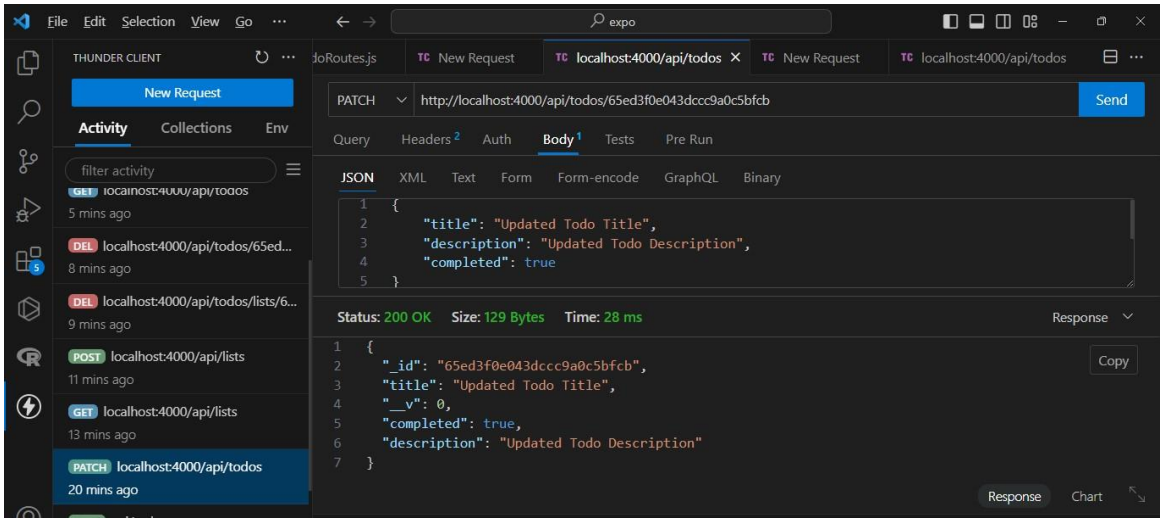
```
// models/todoModel.js

const mongoose = require('mongoose');

const todoSchema = new mongoose.Schema({
  title: String,
  description: String,
  completed: Boolean
});

const Todo = mongoose.model('Todo', todoSchema);

module.exports = Todo;
```

4. OUTPUT:

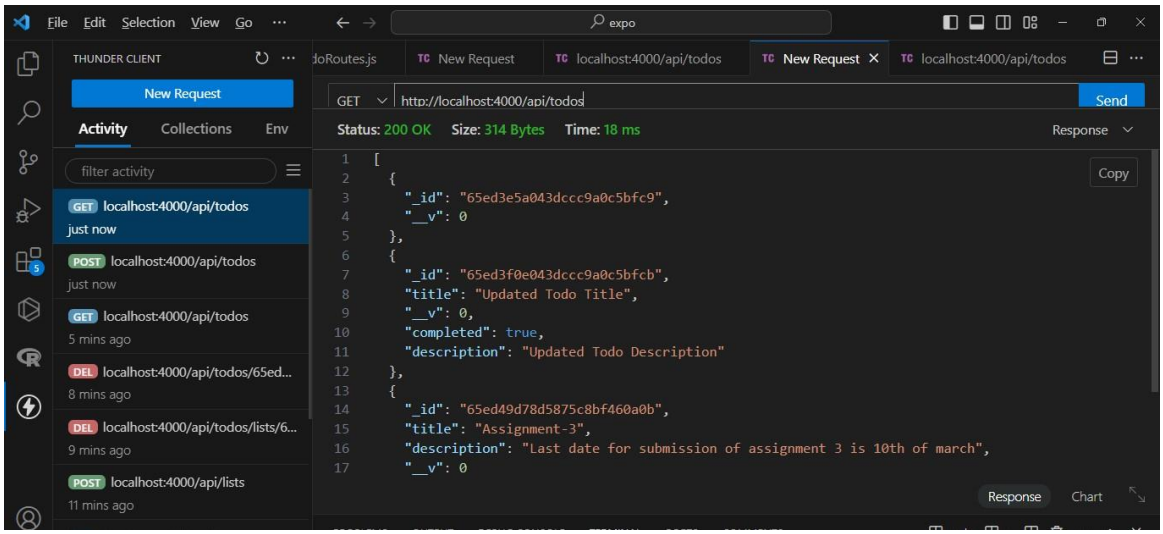
4.1 POST:



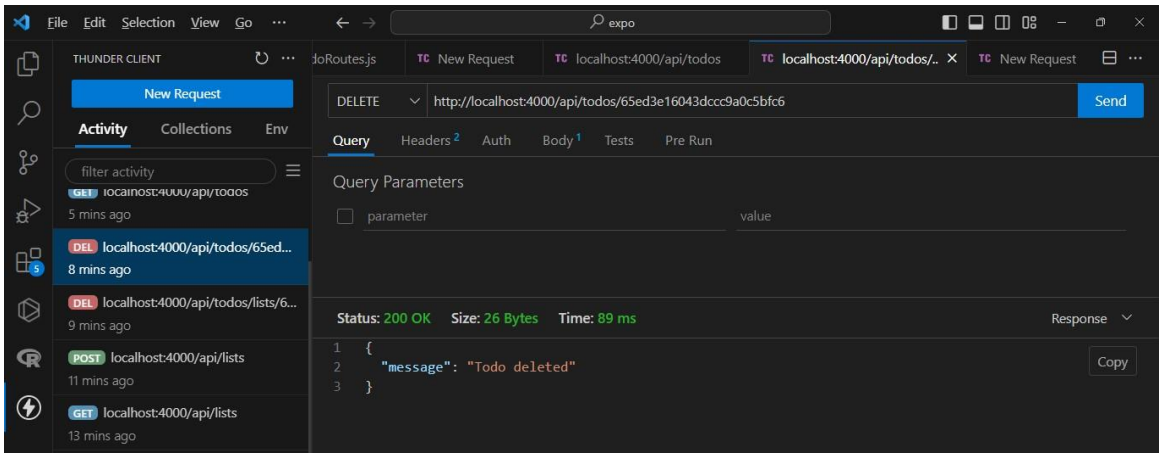
4.2 UPDATE:



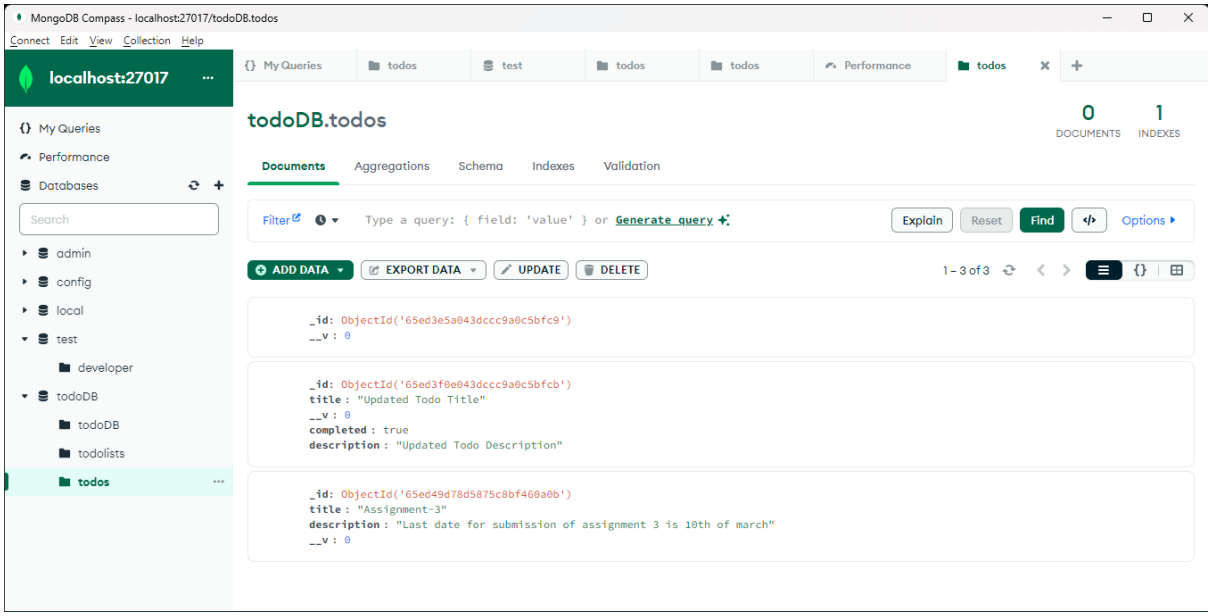
4.3 GET:



4.4 DELETE:



4.5 MONGODB:


MongoDB Compass - localhost:27017/todoDB.todos
Connect Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
admin
config
local
test
developer
todoDB
todoDB
todolists
todos
My Queries
todos
test
todos
todos
Performance
todos
todoDB.todos
0
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
1 - 3 of 3
_id: ObjectId('65ed3e5a043dccc9a0c5bfc9')
__v : 0
_id: ObjectId('65ed3f0e043dccc9a0c5bfcb')
title : "Updated Todo Title"
__v : 0
completed : true
description : "Updated Todo Description"
_id: ObjectId('65ed49d78d5875c8bf460a0b')
title : "Assignment-3"
description : "Last date for submission of assignment 3 is 10th of march"
__v : 0