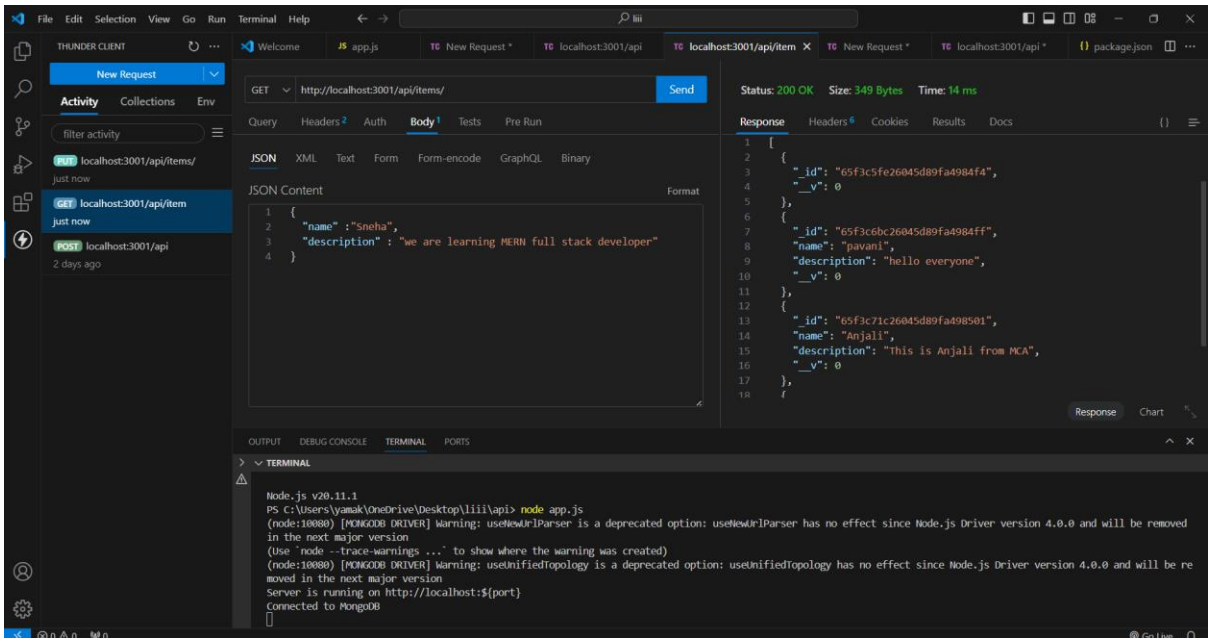
App.js

```
const express = require('express');
const mongoose = require('mongoose');
const app = express();
const port = 3001;
// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/mydatabase', { useNewUrlParser: true,
useUnifiedTopology: true });
const db = mongoose.connection;
db.on('error', console.error.bind(console, 'MongoDB connection error:'));
db.once('open', () => {
 console.log('Connected to MongoDB');
});
// Define a simple model
const Item = mongoose.model('Item', {
 id:Number,
 name: String,
 description: String,
});
// Middleware for parsing JSON
app.use(express.json());
// Routes
app.get('/api/items', async (req, res) => {
 try {
 const items = await Item.find();
 res.json(items);
 } catch (err) {
 res.status(500).json({ error: err.message });
 }
});
app.post('/api/items', async (req, res) => {
```

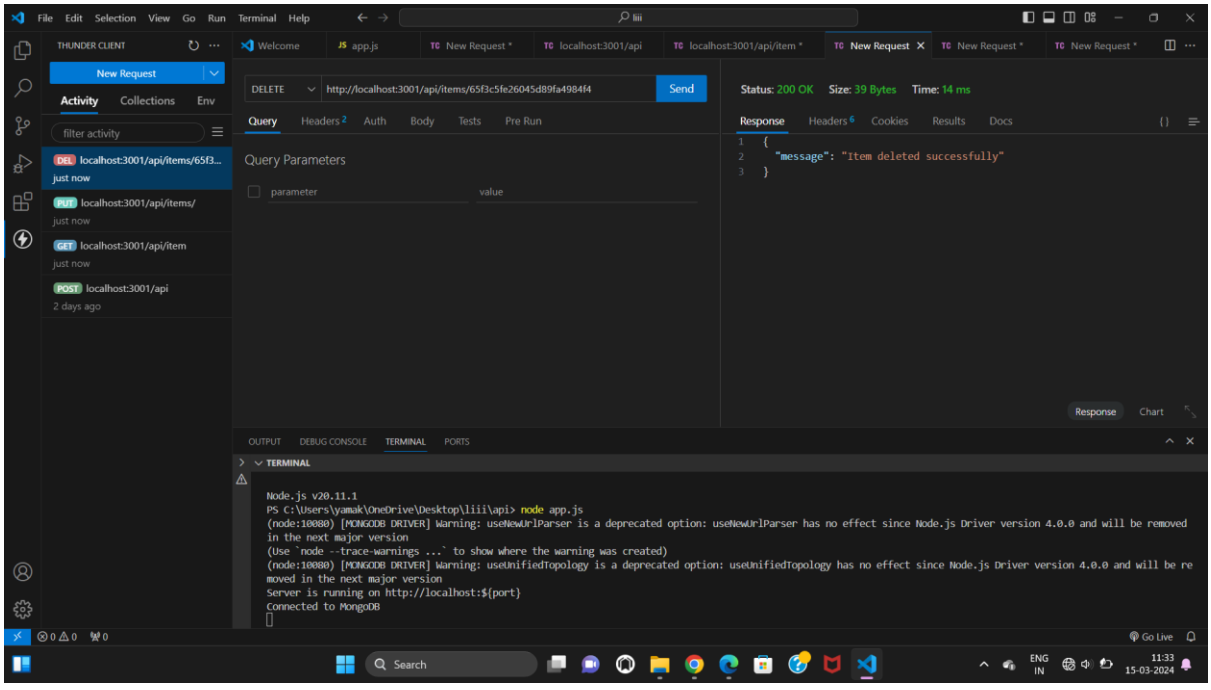
```
 const { name, description } = req.body
try {
 const newItem = new Item({ name, description });
 await newItem.save();
 res.status(201).json(newItem);
 } catch (err) {
 res.status(500).json({ error: err.message });
 }
});
app.put('/api/items/:id', async (req, res) => {
 const { name, description } = req.body;
 const { id } = req.params;
 try {
 const updatedItem = await Item.findByIdAndUpdate(id, { name, description }, { new:
true });
 res.json(updatedItem);
 } catch (err) {
 res.status(500).json({ error: err.message });
 }
});
app.delete('/api/items/:id', async (req, res) => {
 const { id } = req.params;
 try {
 await Item.findByIdAndDelete(id);
 res.json({ message: 'Item deleted successfully' });
 } catch (err) {
 res.status(500).json({ error: err.message });}
});
// Start the server
app.listen(port, () => {
 console.log("Server is running on http://localhost:${port}");});
```

OUTPUT:

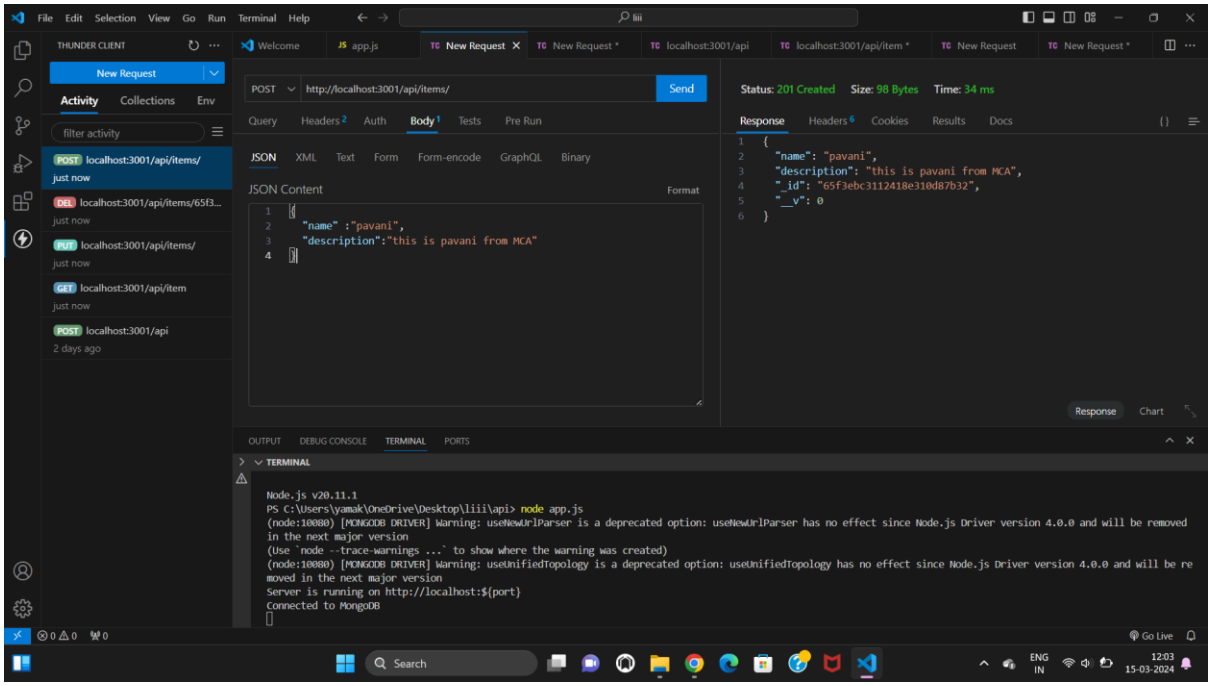
GET METHOD:



DELETE METHOD:



POST METHOD:



PUT METHOD: