server.js

```
const express = require('express'); const mongoose = require('mongoose'); const cors = require('cors');
const todoRoutes = require('./routes/todoRoutes');

const app = express();

// Middleware app.use(cors()); app.use(express.json());

// MongoDB connection (optional — skip if using local array)
mongoose.connect('mongodb://localhost/todoapp')
 .then(() => console.log('✅ MongoDB Connected'))
 .catch(err => console.log(err));
```

```
// Routes

app.use('/api/todos', todoRoutes);


// Start server const PORT = 5000; app.listen(PORT, () => console.log(`🚀 Server running on port
${PORT}`));
```

routes/todoRoutes.js

```
const express = require('express'); const router = express.Router(); const { getTodos, addTodo, deleteTodo
} = require('../controllers/todoController');


// Define routes router.get('/', getTodos); router.post('/', addTodo); router.delete('/:id', deleteTodo);


module.exports = router;
```

controllers/todoController.js

```
const Todo = require('../models/Todo');


// Get all todos exports.getTodos = async (req, res) => {   const todos = await Todo.find();   res.json(todos);

};


// Add a new todo exports.addTodo = async (req, res) => {   const newTodo = new Todo({     task:
req.body.task,     completed: false

  });

  await newTodo.save();   res.json(newTodo);

};
```

```
// Delete a todo exports.deleteTodo = async (req, res) => {   await Todo.findByIdAndDelete(req.params.id);
res.json({ message: 'Todo deleted' });
};
```

models/Todo.js

```
const mongoose = require('mongoose');

const todoSchema = new mongoose.Schema({
 task: String,   completed: { type: Boolean, default: false }
});

module.exports = mongoose.model('Todo', todoSchema);
```

package.json (main parts)

```
{
 "name": "todo-backend",
 "version": "1.0.0",
 "main": "server.js",
 "scripts": {
  "start": "node server.js",
  "dev": "nodemon server.js"
 },
 "dependencies": {
  "cors": "^2.8.5",
  "express": "^4.19.2",
  "mongoose": "^8.0.3"
 },
 "devDependencies": {
    "nodemon": "^3.1.0"
  }
}
```