

App.js:

```
import React, { useState, useEffect } from 'react';
```

```
import axios from 'axios';
```

```
const App = () => {
```

```
  const [todos, setTodos] = useState([]);
```

```
  const [newTodo, setNewTodo] = useState("");
```

```
  useEffect(() => {
```

```
    axios.get('http://localhost:5000/todos')
```

```
      .then(response => setTodos(response.data))
```

```
      .catch(error => console.error(error));
```

```
  }, []);
```

```
  const addTodo = () => {
```

```
    if (!newTodo.trim()) {
```

```
      alert('Todo cannot be empty');
```

```
      return;
```

```
    }
```

```
    axios.post('http://localhost:5000/todos', { text: newTodo })
```

```
      .then(response => {
```

```
        setTodos([...todos, response.data]);
```

```
        setNewTodo("");
```

```
      })
```

```
      .catch(error => console.error(error));
```

```
  };
```

```
  const toggleTodo = (id, completed) => {
```

```
    axios.put(`http://localhost:5000/todos/${id}`, { completed: !completed })
```

```
      .then(() => {
```

```
        setTodos(todos.map(todo => todo.id === id ? { ...todo, completed: !completed } : todo));
```

```
    })
    .catch(error => console.error(error));
};
```

```
const deleteTodo = (id) => {
  axios.delete(`http://localhost:5000/todos/${id}`)
    .then(() => {
      setTodos(todos.filter(todo => todo.id !== id));
    })
    .catch(error => console.error(error));
};
```

```
return (
  <div>
    <h1>Todo App</h1>
    <input
      type="text"
      value={newTodo}
      onChange={(e) => setNewTodo(e.target.value)}
      placeholder="Add a new todo"
    />
    <button onClick={addTodo}>Add</button>
    <ul>
      {todos.map(todo => (
        <li key={todo.id}>
          <span
            style={{ textDecoration: todo.completed ? 'line-through' : 'none' }}
            onClick={() => toggleTodo(todo.id, todo.completed)}
          >
            {todo.text}
          </span>

```

```
      <button onClick={() => deleteTodo(todo.id)}>Delete</button>
    </li>
  )}
</ul>
</div>
);
};
```

```
export default App;
```

```
server.js: (expressJs)
```

```
const express = require('express');
const cors = require('cors');
```

```
const app = express();
const PORT = 5000;
```

```
app.use(cors());
app.use(express.json());
```

```
let todos = [];
```

```
// Routes
```

```
app.get('/todos', (req, res) => {
  res.json(todos);
});
```

```
app.post('/todos', (req, res) => {
  const { text } = req.body;
  const newTodo = { id: Date.now(), text, completed: false };
  todos.push(newTodo);
```

```
    res.status(201).json(newTodo);
  });

  app.put('/todos/:id', (req, res) => {
    const { id } = req.params;
    const { completed } = req.body;

    const todoExists = todos.some(todo => todo.id === parseInt(id));
    if (!todoExists) {
      return res.status(404).json({ message: 'Todo not found' });
    }

    todos = todos.map(todo =>
      todo.id === parseInt(id) ? { ...todo, completed } : todo
    );
    res.json({ message: 'Todo updated successfully' });
  });

  app.delete('/todos/:id', (req, res) => {
    const { id } = req.params;
    todos = todos.filter(todo => todo.id !== parseInt(id));
    res.json({ message: 'Todo deleted successfully' });
  });

  app.listen(PORT, () => {
    console.log(` Server running on http://localhost:${PORT} `);
  });
```

