Priyanshu Dhyani

A simple "Todo List" web application
(frontend using React and backend using Node.js/Express)

1. Setting Up the Backend (Node.js/Express)

Step 1: Initialize the Project

```
mkdir todo-list
cd todo-list
npm init -y
```

Step 2: Install Dependencies

```
npm install express mongoose cors body-parser
npm install nodemon --save-dev
```

Step 3: Create the Server

Create a file named server.js and add the following code:

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

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

app.use(cors());
app.use(bodyParser.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/todo', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

// Define Todo model
const Todo = mongoose.model('Todo', new mongoose.Schema({
  title: String,
  completed: Boolean,
}));

// Routes
app.get('/todos', async (req, res) => {
  const todos = await Todo.find();
```

```
 res.json(todos);
});

app.post('/todos', async (req, res) => {
  const newTodo = new Todo({
    title: req.body.title,
    completed: false,
  });
  const savedTodo = await newTodo.save();
  res.json(savedTodo);
});

app.put('/todos/:id', async (req, res) => {
  const updatedTodo = await Todo.findByIdAndUpdate(req.params.id, req.body, { new: true });
  res.json(updatedTodo);
});

app.delete('/todos/:id', async (req, res) => {
  await Todo.findByIdAndDelete(req.params.id);
  res.json({ message: 'Todo deleted' });
});

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

Step 4: Configure package.json

Add the following to your scripts section in package.json to use nodemon for auto-reloading:

json

```
"scripts": {
  "start": "node server.js",
  "dev": "nodemon server.js"
}
```

Step 5: Run the Server

npm run dev

2. Setting Up the Frontend (React)

Step 1: Create React App

npx create-react-app client

cd client

Priyanshu Dhyani

Step 2: Install Axios
npm install axios

Step 3: Create Components
To enhance the look and feel update App.css file:
App.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f5f5f5;
  margin: 0;
  padding: 0;
}

.App {
  max-width: 600px;
  margin: 50px auto;
  padding: 20px;
  background: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
}

h1 {
  text-align: center;
  color: #333;
}

form {
  display: flex;
  justify-content: space-between;
  margin-bottom: 20px;
}

input[type="text"] {
  flex: 1;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
  margin-right: 10px;
}
```

Priyanshu Dhyani

```
button {
 padding: 10px 20px;
  border: none;
  background-color: #28a745;
  color: #fff;

 border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #218838;
}

.todo-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px;
  border-bottom: 1px solid #eee;
}

.todo-item span {
  flex: 1;
}

.todo-item button {
  margin-left: 10px;
  background-color: #dc3545;
}

.todo-item button:hover {
  background-color: #c82333;
}

.todo-item button.complete {
  background-color: #ffc107;
}

.todo-item button.complete:hover {
  background-color: #e0a800;
}
```

```
.todo-item span.completed {
  text-decoration: line-through;
  color: #aaa;
}
```

In the src directory, create a folder named components and add the following files:

- TodoList.js
- TodoItem.js
- AddTodo.js

AddTodo.js

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

const AddTodo = ({ fetchTodos }) => {
  const [title, setTitle] = useState('');

  const handleSubmit = async (e) => {
    e.preventDefault();
    if (!title) return;

    await axios.post('http://localhost:5000/todos', { title });
    setTitle('');
    fetchTodos();
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        placeholder="Add a new todo"
        value={title}
        onChange={(e) => setTitle(e.target.value)}
      />
      <button type="submit">Add</button>
    </form>
  );
};

export default AddTodo;
```

Priyanshu Dhyani

TodoItem.js

```
import React from 'react';
import axios from 'axios';

const TodoItem = ({ todo, fetchTodos }) => {
  const toggleComplete = async () => {
    await axios.put(`http://localhost:5000/todos/${todo._id}`, {
      completed: !todo.completed,
    });
    fetchTodos();
  };

  const deleteTodo = async () => {
    await axios.delete(`http://localhost:5000/todos/${todo._id}`);
    fetchTodos();
  };

  return (
    <div className="todo-item">
      <span className={todo.completed ? 'completed' : ''}>
        {todo.title}
      </span>
      <button className="complete" onClick={toggleComplete}>
        {todo.completed ? 'Undo' : 'Complete'}
      </button>
      <button onClick={deleteTodo}>Delete</button>
    </div>
  );
};

export default TodoItem;
```

TodoList.js

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import TodoItem from './TodoItem';
import AddTodo from './AddTodo';
const TodoList = () => {
  const [todos, setTodos] = useState([]);
  const fetchTodos = async () => {
    const response = await axios.get('http://localhost:5000/todos');
  setTodos(response.data);
  };
```

```
  useEffect(() => {
    fetchTodos();
  }, []);
  return (
    <div>
      <h1>Todo List</h1>
      <AddTodo fetchTodos={fetchTodos} />
      {todos.map((todo) => (
        <TodoItem key={todo._id} todo={todo} fetchTodos={fetchTodos} />

 ))}
    </div>
  );
};
export default TodoList;
```

Step 4: Update App.js
Replace the content of App.js with:

```
import React from 'react';
import TodoList from './components/TodoList';
import './App.css';

const App = () => (
  <div className="App">
    <TodoList />
  </div>
);
export default App;
```

Step 5: Run the React App

```
npm start
```

Priyanshu Dhyani

Screenshots

Frontend:

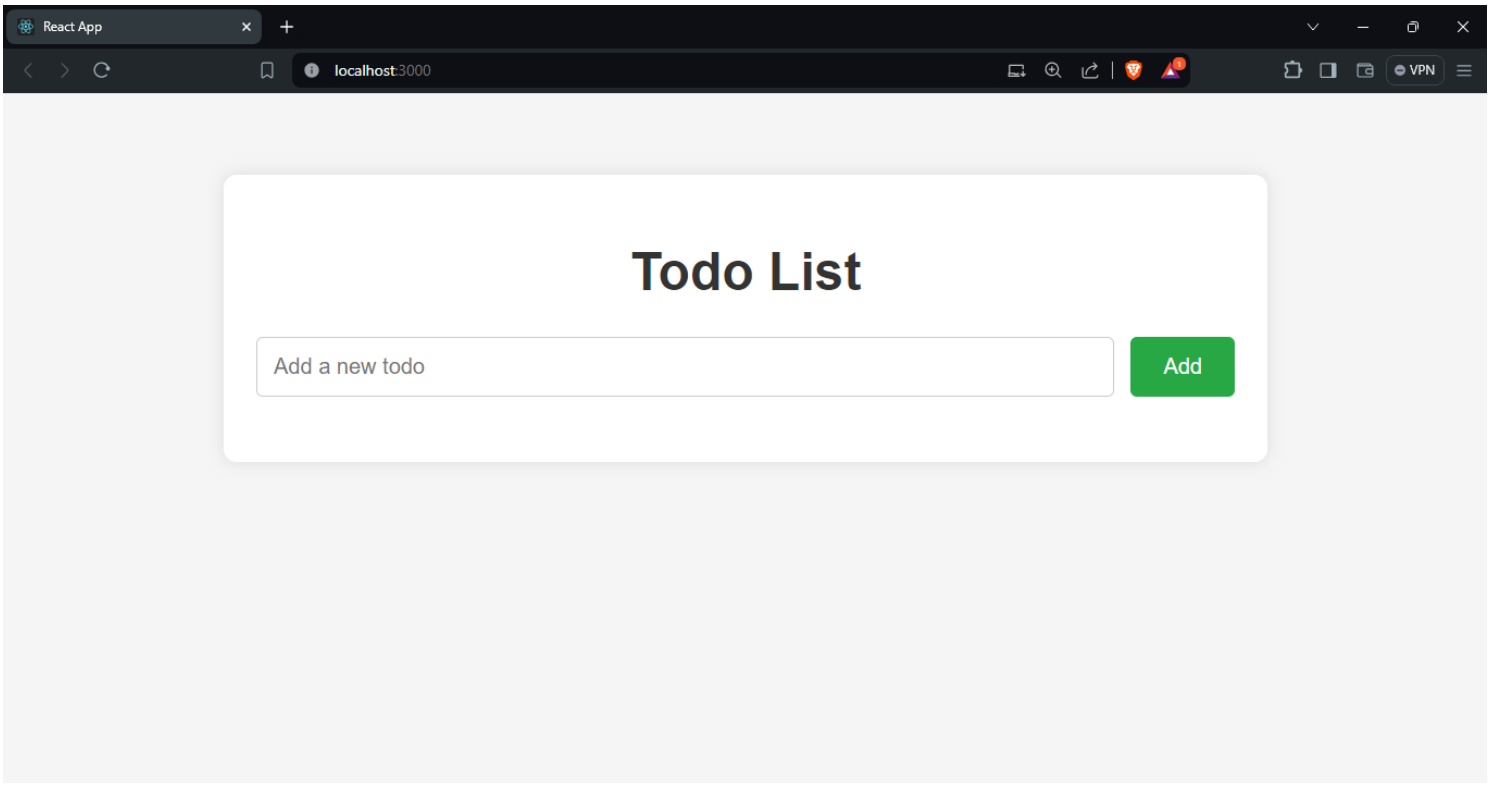

Fig 1. (front page)

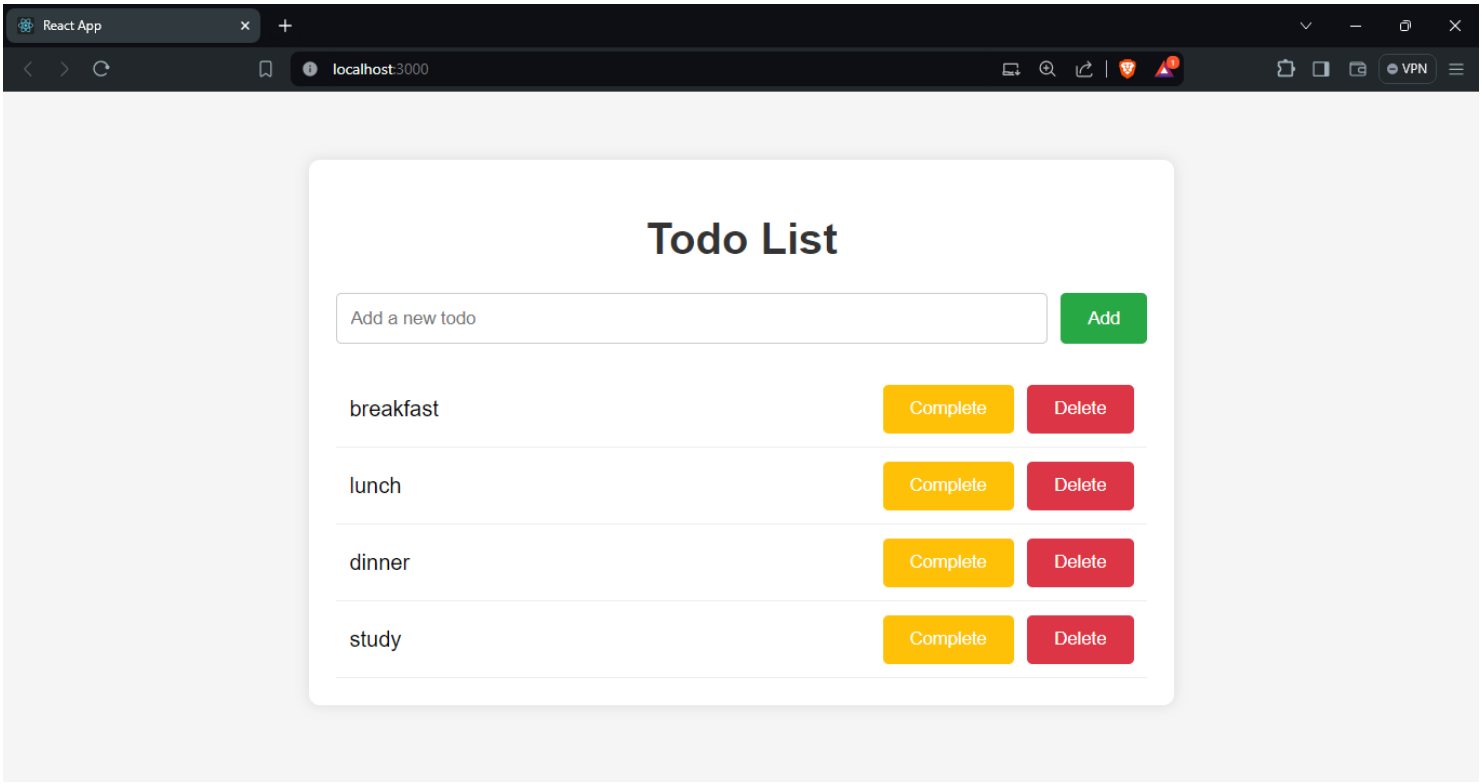

Fig 2. (adding data)

Screenshots

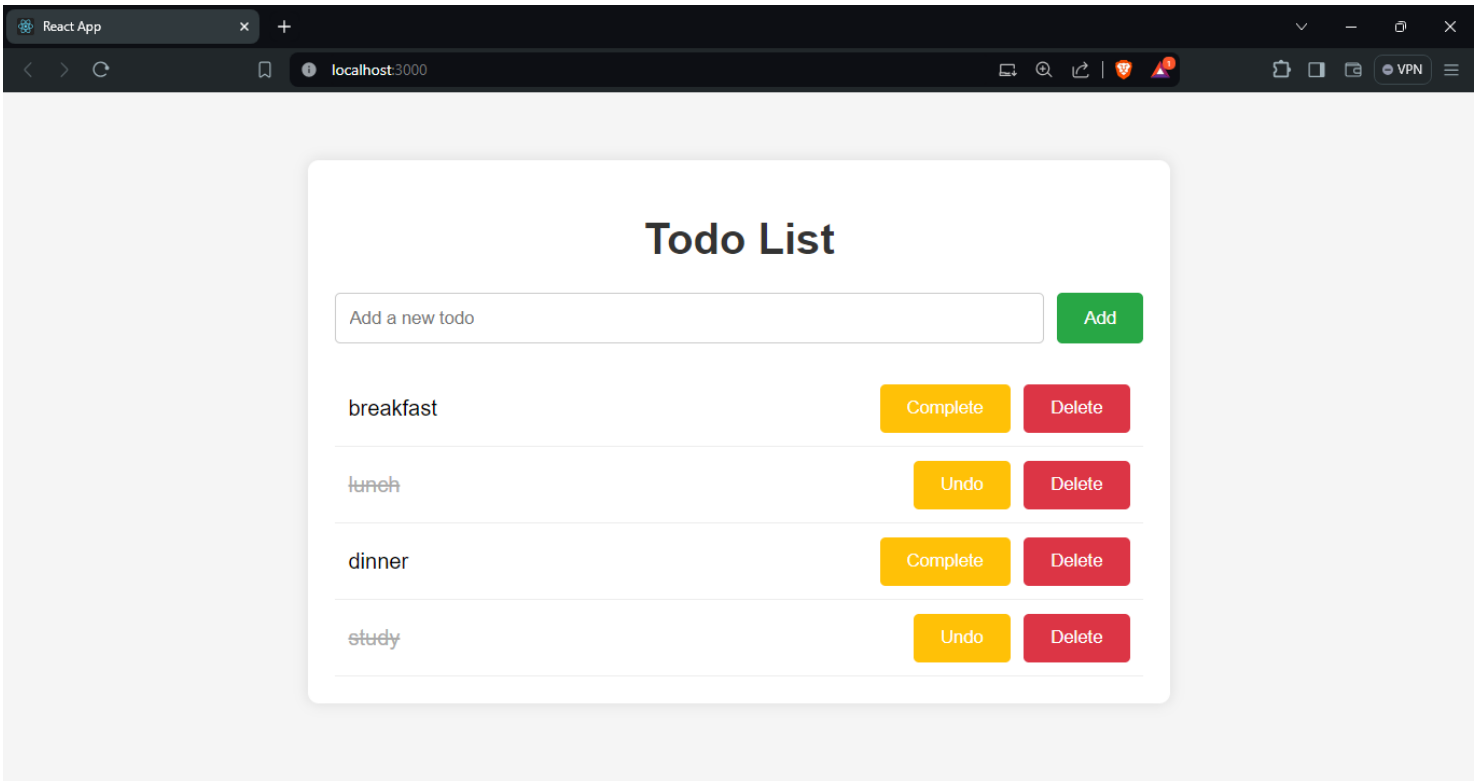

Fig 3. (Responses)

Backend:

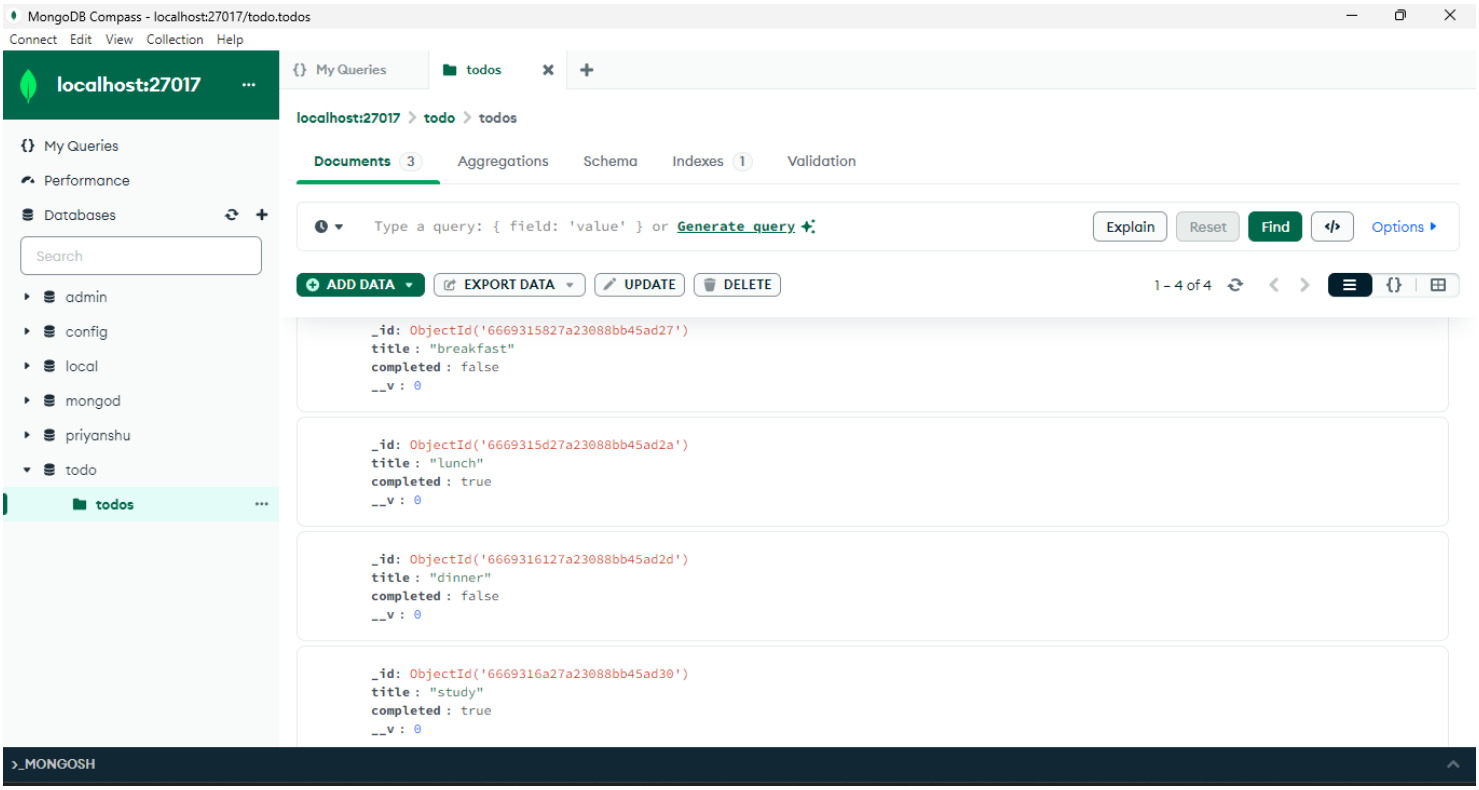

Fig 4. (DataBase - MongoDB)