MERN Stack To-Do List App

MERN Stack To-Do List App

Backend (Node.js + Express + MongoDB)

Install Dependencies
```sh
npm install express mongoose cors dotenv
```

server.js
```javascript
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
require('dotenv').config();

const app = express();
app.use(express.json());
app.use(cors());

mongoose.connect(process.env.MONGO_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
});

const TodoSchema = new mongoose.Schema({
    task: String,
    completed: Boolean
});

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

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

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

app.put('/todos/:id', async (req, res) => {
    await Todo.findByIdAndUpdate(req.params.id, req.body);
    res.json({ message: 'Updated successfully' });
});

app.delete('/todos/:id', async (req, res) => {
```

```javascript
    await Todo.findByIdAndDelete(req.params.id);
    res.json({ message: 'Deleted successfully' });
});


app.listen(5000, () => console.log('Server running on port 5000'));
```


Frontend (React.js)

Install React App & Dependencies
```sh
npx create-react-app client
cd client
npm install axios
```


App.js
```javascript
import React, { useState, useEffect } from 'react';
import axios from 'axios';

function App() {
    const [todos, setTodos] = useState([]);
    const [task, setTask] = useState('');

    useEffect(() => {
        axios.get('http://localhost:5000/todos').then(res => setTodos(res.data));
    }, []);

    const addTodo = () => {
        axios.post('http://localhost:5000/todos', { task, completed: false })
            .then(res => setTodos([...todos, res.data]));
    };

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

    return (
        <div>
            <h1>To-Do List</h1>
            <input value={task} onChange={(e) => setTask(e.target.value)} />
            <button onClick={addTodo}>Add</button>
            <ul>
                {todos.map(todo => (
                    <li key={todo._id}>
                        {todo.task} <button onClick={() => deleteTodo(todo._id)}>Delete</button>
                    </li>
                ))}
            </ul>
        </div>
    );
}
```

```
export default App;
```

Running the Application

Start Backend
```sh
node server.js
```

Start Frontend
```sh
cd client
npm start
```

Upload to GitHub
```sh
git init
git add .
git commit -m "Initial commit"
git remote add origin <GitHub_Repo_URL>
git push -u origin main
```