Here's a step-by-step guide to **integrating the front-end and back-end** for the task management application  
  
**Step 1: Set Up the Back-End**

**1. Install Node.js**

* Download and install Node.js from <https://nodejs.org>.
* Verify installation:

Type following command in the VS code terminal

node -v

npm -v

**2. Initialize the Back-End Project**

1. Create a project folder and navigate to it:

mkdir task-manager-backend

cd task-manager-backend

1. Initialize the project with npm:

npm init -y

* + This generates a package.json file to manage dependencies.

1. **Install Required Packages**

**Install mongodb using below link**

[**https://downloads.mongodb.com/compass/mongodb-compass-1.45.0-win32-x64.exe**](https://downloads.mongodb.com/compass/mongodb-compass-1.45.0-win32-x64.exe)

Install Express.js for building APIs and Mongoose for MongoDB interaction:

npm install express mongoose cors

**4. Create the Back-End Structure**

1. Create the server.js file:

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(express.json());

mongoose.connect('mongodb://localhost:27017/taskdb', {

useNewUrlParser: true,

useUnifiedTopology: true,

}).then(() => console.log('MongoDB connected'))

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

// Task Schema and Model

const taskSchema = new mongoose.Schema({

title: String,

description: String,

completed: Boolean,

});

const Task = mongoose.model('Task', taskSchema);

// API Endpoints

app.get('/tasks', async (req, res) => {

const tasks = await Task.find();

res.json(tasks);

});

app.post('/tasks', async (req, res) => {

const newTask = new Task(req.body);

await newTask.save();

res.json(newTask);

});

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

1. Start the server:

node server.js

* + Open http://localhost:5000/tasks in your browser.
  + You should see an empty array ([]) if no tasks exist.

**Step 2: Set Up the Front-End**

**1. Create the Front-End Project**

1. Navigate to a directory for the front-end:

mkdir task-manager-frontend

cd task-manager-frontend

1. Create a React app:

npx create-react-app .

**2. Install Axios**

Axios is used for making API calls:

npm install axios

**3. Create a Simple Task Manager**

1. Edit src/App.js:

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

import axios from 'axios';

const App = () => {

const [tasks, setTasks] = useState([]);

const [newTask, setNewTask] = useState({ title: '', description: '', completed: false });

useEffect(() => {

axios.get('http://localhost:5000/tasks')

.then(response => setTasks(response.data))

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

}, []);

const handleAddTask = () => {

axios.post('http://localhost:5000/tasks', newTask)

.then(response => setTasks([...tasks, response.data]))

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

};

return (

<div>

<h1>Task Manager</h1>

<ul>

{tasks.map(task => (

<li key={task.\_id}>{task.title} - {task.description}</li>

))}

</ul>

<input

type="text"

placeholder="Title"

value={newTask.title}

onChange={(e) => setNewTask({ ...newTask, title: e.target.value })}

/>

<input

type="text"

placeholder="Description"

value={newTask.description}

onChange={(e) => setNewTask({ ...newTask, description: e.target.value })}

/>

<button onClick={handleAddTask}>Add Task</button>

</div>

);

};

export default App;

1. Start the React app:

npm start

* + Open http://localhost:3000 to see the front-end.

**Step 3: Integrate the Front-End and Back-End**

1. **Test API Integration**:
   * Add a task using the front-end form.
   * Verify that the task appears in the task list (from the API response).
2. **Cross-Origin Requests (CORS)**:
   * The back-end has cors enabled, allowing the React app to communicate with it.
3. **Connect Database**:
   * Ensure MongoDB is running locally. Run:

mongod

* + Check data in MongoDB:

mongo





