TaskForm.jsx

import React, { useState } from 'react';

const TaskForm = ({ addTask }) => {

  const [title, setTitle] = useState('');

  const [description, setDescription] = useState('');

  const handleSubmit = (e) => {

    e.preventDefault();

    if (title && description) {

      const newTask = { title, description };

      addTask(newTask);

      setTitle('');

      setDescription('');

    }

  };

  return (

    <form onSubmit={handleSubmit} className="task-form">

      <input

        type="text"

        placeholder="Task Title"

        value={title}

        onChange={(e) => setTitle(e.target.value)}

        required

      />

      <textarea

        placeholder="Task Description"

        value={description}

        onChange={(e) => setDescription(e.target.value)}

        required

      />

      <button type="submit">Add Task</button>

    </form>

  );

};

export default TaskForm;

TaskList.jsx

import React from 'react';

const TaskList = ({ tasks, deleteTask }) => {

  return (

    <div className="task-list">

      {tasks.length > 0 ? (

        tasks.map(task => (

          <div key={task.id} className="task-item">

            <h3>{task.title}</h3>

            <p>{task.description}</p>

            <button onClick={() => deleteTask(task.id)}>Delete</button>

          </div>

        ))

      ) : (

        <p>No tasks available</p>

      )}

    </div>

  );

};

export default TaskList;

App.jsx

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

import axios from 'axios';

import TaskList from './COMPONENTS/TaskList';

import TaskForm from './COMPONENTS/TaskForm';

import './App.css';

const App = () => {

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

  // Fetch tasks from JSON Server on initial load

  useEffect(() => {

    axios.get('http://localhost:3000/data')

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

      .catch(error => console.error('Error fetching tasks:', error));

  }, []);

  // Add a new task

  const addTask = (task) => {

    axios.post('http://localhost:3000/data', task)

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

      .catch(error => console.error('Error adding task:', error));

  };

  // Delete a task

  const deleteTask = (id) => {

    axios.delete(`http://localhost:3000/data/${id}`)

      .then(() => setTasks(tasks.filter(task => task.id !== id)))

      .catch(error => console.error('Error deleting task:', error));

  };

  return (

    <div className="app">

      <h1>Task Manager</h1>

      <TaskForm addTask={addTask} />

      <TaskList tasks={tasks} deleteTask={deleteTask} />

    </div>

  );

};

export default App;

allAPI.js

import axios from 'axios';

const API\_URL = 'http://localhost:3000/tasks'; // JSON Server URL

// Get all tasks

export const getTasks = async () => {

  try {

    const response = await axios.get(API\_URL);

    return response.data;

  } catch (error) {

    console.error('Error fetching tasks:', error);

    throw error;

  }

};

// Add a new task

export const addTask = async (task) => {

  try {

    const response = await axios.post(API\_URL, task);

    return response.data;

  } catch (error) {

    console.error('Error adding task:', error);

    throw error;

  }

};

// Delete a task by ID

export const deleteTask = async (id) => {

  try {

    await axios.delete(`${API\_URL}/${id}`);

  } catch (error) {

    console.error('Error deleting task:', error);

    throw error;

  }

};

commonAPI.js

import axios from 'axios';

export const commonAPI = async (httpMethod, url, reqBody) => {

    let reqConfig = {

        method: httpMethod,  // Make sure httpMethod is passed here

        url,

        data: reqBody

    };

    return await axios(reqConfig).then((response) => {

            return response;

        })

        .catch((error) => {

            return error;

        });

};

serverURL.js

export const serverUrl='http://localhost:3000'

db.json

{

    "data":[]

}

Requirement

React Project: Task Manager

Objective: Build a Task Manager app using React and JSON Server.

Folder Structure:

components/ and pages/ folders in your src/ directory.

JSON Server Setup:

Use JSON Server for storing and fetching tasks with fields: id, title, description.

Components:

App: Main component.

TaskList: Displays tasks with delete buttons.

TaskForm: Form to add new tasks.

Functionality:

Add Task: Form to submit new tasks.

View Tasks: Display tasks from JSON Server.

Delete Task: Remove tasks from JSON Server.

State Lifting: Manage tasks state in App, pass to TaskList and TaskForm.

Nested Components: Implement nested components for task display and form.

API Fetching:

Use axios or Fetch API for tasks operations.

Deliverables:

React app with listed functionalities.

Instructions for running JSON Server and the React app.