

Assignment Solution- Task Manager App

1. Backend Task Manager App Solution

- CRUD Operations:
 - Users: Implement API endpoints for Create, Read, Update, Delete using user schema (ID, name, email).
 - Projects: CRUD endpoints with project schema including ownerId, project name, and shared users with roles.
 - Tasks: CRUD for tasks with support for nested sub-tasks using parentTaskId.
- Recursive Sub-tasks:
 - Design tasks table with a 'parentTaskId' field referencing the same table.
 - Recursive query or tree-traversal functions fetch sub-tasks for a parent task.
- Filter Tasks:
 - Implement query parameters for filtering tasks based on 'status', 'dueDate', or keyword in 'name' or 'description'.
- Role-Based Access:
 - Define roles (Owner, Editor, Viewer) at project-user level.
 - Apply middleware logic to enforce permissions on endpoints based on assigned role.

2. Frontend Task Manager Dashboard Solution

- Authentication:
 - Use a mock login for 'User A'. Store user in React Context or Vuex store.
- Users Section:
 - List all users.
 - Allow add/edit/delete operations using modals or forms.
- Projects Section:
 - List projects owned or shared with User A.
 - Allow creating, updating, deleting, and sharing projects with role assignments.
 - Display user role badge next to each project.
- Tasks Section:
 - Display tasks in collapsible tree structure using recursion.
 - Each task includes name, description, due date, status, parentTaskId.
 - Add/Edit/Delete task functionality with modal forms.
 - Filters: Implement dropdowns or date-pickers for status, due date range, and search input.
- Role-Based Access (Simulated):
 - Simulate access control per project:
 - Owner – Full access

- Editor – Modify tasks, restricted sharing
- Viewer – Read-only

3. Bonus Features

- Drag-and-drop using libraries like React DnD.
- Color indicators: Red (Overdue), Yellow (In Progress), Green (Done).
- Save user filters using localStorage.

4. Mock Data

- Use provided mock JSON data to simulate user, project, and task data.
- Fetch data using Fetch API in frontend.

5. Technology Stack

- React with Context API Vuex.
- Tree rendering with recursion for sub-task hierarchy.
- Bootstrap, Tailwind, or Material UI for styling.
- React-Datepicker or equivalent for date input.
- Axios for API calls.