Streamframe Coding Assignment

Prepared by: Sergio M. Lio (applicant)

Tech stack: javascript, reactjs, bootstrap 5, node express, mysql, sequelize.

Requirement list:

* ✔Create a task listing page that shows a flat list of individual tasks and: o Each task’s ID, description, and status.
* ✔A status checkbox for each task to toggle its status between “**DONE”** and **“IN PROGRESS”** when clicked.
* ✔The task listing page should feature a status filter (IN PROGRESS, DONE, COMPLETE).
* ✔Create a task creation form that takes the following inputs:

- Task Name (required).

- Parent Task ID (optional).

* ✔Check for and prevent circular dependencies when creating a task that specifies a parent.
* ✔Upgrade the task listing page so that parent tasks also show:

- The total number of dependencies.

- The number of dependencies marked as DONE.

- The number of dependencies marked as COMPLETE.

* ✔Upgrade the task listing page so that parent tasks also show:

i- Marking a task as DONE will also check the status of all dependencies. When all dependencies are COMPLETE, mark the task as COMPLETE (instead of DONE).

✔- Marking a task as IN PROGRESS (by clearing the status checkbox) should update its parent task (if it has one) so that the parent’s status changes from COMPLETE to DONE. A parent task must not revert to IN PROGRESS from DONE or COMPLETE.

i- Repeat these two processes on the task’s parents (if any) until the status change has propagated all the way to the top of the hierarchy.

* ✔Upgrade the task listing page from a flat list to a nested hierarchical list. That is, all of the dependencies for a task should appear in a separate list within the parent task.
* ✔Upgrade the task listing page to allow tasks to be edited:

✔- Allow a task’s name to be changed.

i- Allow a task’s parent task to be changed. Doing this must trigger a status change propagation behaviour as described above.

API Routes Endpoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| method | url | params | Form-body | data |
| get | /api/task | - | - | retrieve all the task including subtask |
| get | /api/subtask/:id | id | - | retrieve all the subtask with taskId |
| post | /api/task | - | name, status | Create new task |
| post | /api/subtask | - | title, status, taskId | Create new subtask |
| put | /api/task/:id | id | name, status | Update task |
| put | /api/subtask/:id | id | name, status | Update subtask |

Installation: backend

Clone the github repo: on cli via https

`git clone <https://github.com/skitband/Streamframe-Coding-Assignment.git`>

`cd Streamframe-Coding-Assignment`

`cd backend`

`npm install`

Create Database from your localhost mysql with db name: **taskapp**

`Refer to db.config.js file in backend/config, and Update on your database base on your db configuration (refer on the example down)`

`module.exports = {

  HOST: "localhost",

  USER: "root",

  PASSWORD: "password",

  DB: "taskapp",

  dialect: "mysql",

  pool: {

    max: 5,

    min: 0,

    acquire: 30000,

    idle: 10000

  }

};`

Run backend server

`nodemon server.js`

The server will create models for you.

Open [http://localhost:8080](http://localhost:8080) to view it in your browser.

Installation: frontend

`cd client`

`npm install`

`npm install`

Runs the app in the development mode.\

Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

**Done**!