NYI - Full-Stack Kanban Task Board Assignment

Project Overview:

Build a simple Kanban-style task board. The application allows users to view tasks, and only the task creator (or an admin) can modify their tasks.

Backend (Node.js, Express, MongoDB)

Task API:

- **Data Model (Task):**
  - **`id`:** Unique identifier
  - **`title`:** String
  - **`description`:** String
  - **`status`:** Enum: `"To Do"`, `"In Progress"`, `"Done"`
  - **`position`:** Number (ordering in column)
  - **`createdBy`:** Reference to the user who created the task

Endpoints:

- **`POST /tasks`**: Create a new task (authentication required)
- **`GET /tasks`**: Get all tasks (accessible to all users)
- **`PUT /tasks/:id`**: Update task (only creator or admin)
- **`DELETE /tasks/:id`**: Delete task (only creator or admin)

Authentication & Authorization:

- Use **JWT** for login/registration.
- **User Roles:**
  - **`user`:** Can only manage their own tasks.
  - **`admin`:** Can manage all tasks.
- Simple role-based access control for task-modifying endpoints.

Frontend (React)

User Interface:

- Display tasks in a **Kanban board** layout with the following columns:
  - **`"To Do"`**, **`"In Progress"`**, **`"Done"`**
- Tasks should show who created them (either by name or username).

Drag-and-Drop:

- Allow **dragging and dropping tasks** within columns.
- Users can only drag their own tasks (unless they are an admin).
- Non-draggable tasks for users who cannot edit them.

Authentication:

- A simple **login/register** form with JWT authentication.
- Store JWT in **`localStorage`** and attach it to API requests.
- UI should reflect user roles (admin sees full controls, regular users only see/edit their tasks).

Optional Features - Good to have (for Enhanced Functionality):

Socket.IO (Real-Time Updates):

- Add **Socket.IO** for real-time updates across clients:
  - Broadcast task creation, updates, and deletions to all connected users.
  - **Real-time feedback** when someone moves a task between columns.

Key Focus Areas:

- **Backend**:
  - Task management (CRUD).
  - Basic role-based authentication (JWT, roles: user/admin).
- **Frontend**:
  - Kanban board layout (To Do, In Progress, Done).
  - Drag-and-drop functionality using **`react-beautiful-dnd`**.
  - Basic user login/register flow with JWT.

Deliverables:

- **Git Repository** containing:
  - **`/server`:** Express backend.
  - **`/client`:** React frontend.
- **`README.md`:** Full setup instructions.
- **`.env`** file for sensitive environment variables (e.g., database URL, JWT secret).

Simplified Flow:

- **Backend**: The server will handle task management and user authentication. Task data will be fetched via REST API **(`GET /tasks`)**, and users can create **(`POST /tasks`)**, update (**`PUT /tasks/:id`**), or delete **(`DELETE /tasks/:id`)** their tasks.
- **Frontend**: The client will display the tasks in the Kanban layout. Users can drag and drop tasks between columns, but only their own tasks can be modified unless the user is an admin. Authentication is handled with JWT stored in **`localStorage`**.