Dưới đây là tổng quan cập nhật về chức năng "Nhóm" (Teams) của bạn, có tính đến tất cả các bổ sung mới nhất:

Tổng quan Chức năng "Nhóm" (Teams) (Cập nhật)

Mục tiêu: Cho phép người dùng (Admin) tạo, quản lý các nhóm làm việc, thêm thành viên vào nhóm và liên kết nhóm với các dự án (hiện có hoặc mới tạo nhanh), với phân quyền rõ ràng trong nhóm.

I. Cấu trúc và Mối quan hệ (Dựa trên ERD và các bảng đã có):

USERS: Chứa thông tin người dùng.

TEAMS: Quản lý thông tin về nhóm.

Fields chính: \_id, team\_name, description, created\_by (liên kết đến USERS.\_id), created\_at, updated\_at.

Thêm is\_deleted (Boolean, default false) để hỗ trợ soft delete nhóm (đề xuất mới).

TEAM\_MEMBERS: Quản lý thành viên trong nhóm và vai trò của họ.

Fields chính: \_id, team\_id (FK đến TEAMS.\_id), user\_id (FK đến USERS.\_id), role (String), joined\_at.

Bổ sung role: ENUM('Admin', 'Editor', 'Member'), DEFAULT 'Member'.

Mặc định: Người tạo nhóm (created\_by trong TEAMS) sẽ tự động được thêm vào TEAM\_MEMBERS với role: 'Admin'.

is\_active (Boolean, default true), approval\_status (Boolean) - giữ nguyên để quản lý trạng thái thành viên và quy trình duyệt (nếu có).

UNIQUE(team\_id, user\_id): Đảm bảo mỗi user chỉ là thành viên một lần trong mỗi nhóm.

PROJECTS: Chứa thông tin dự án.

Fields chính: \_id, project\_name, description, start\_date, end\_date, status, priority, project\_type, is\_deleted, v.v.

Để liên kết với Nhóm: Thêm trường team\_id (ObjectId, nullable) vào model PROJECTS. Điều này giả định một dự án có thể thuộc về một nhóm duy nhất (hoặc không thuộc nhóm nào).

PROJECT\_MEMBERS: Quản lý thành viên trong dự án.

II. Phân quyền vai trò trong nhóm (Quan trọng):

Admin (Duy nhất):

Là người tạo nhóm (created\_by).

Quyền tối cao (CRUD tối cao): Có thể thực hiện mọi thao tác:

Tạo, đọc, cập nhật, xóa nhóm.

Thêm, sửa vai trò (Admin/Editor/Member), xóa thành viên khỏi nhóm.

Tính duy nhất: Trong một nhóm, chỉ có duy nhất một Admin. Khi tạo nhóm, người tạo sẽ mặc định là Admin.

Không thể tự xóa/sửa: Admin không thể tự xóa hoặc tự hạ vai trò của chính mình. Để thay đổi Admin, cần một Admin khác hoặc quy trình chuyển quyền hạn đặc biệt.

Editor:

Quyền: Có thể đọc danh sách thành viên nhóm, thêm thành viên mới, sửa vai trò của Member, và xóa Member.

Hạn chế:

Không thể sửa vai trò của Admin.

Không thể xóa Admin.

Không thể xóa Editor khác (chỉ Admin mới có thể).

Không có quyền xóa hoặc sửa thông tin nhóm (team\_name, description).

Member:

Quyền: Chỉ có quyền đọc thông tin nhóm và danh sách thành viên trong nhóm.

Không thể thêm, sửa, xóa thành viên hay vai trò.

Không thể sửa thông tin nhóm.

III. Giao diện người dùng (Frontend):

Trang "Nhóm" chính (/teams):

Nút "Tạo Nhóm Mới": Mở modal CreateTeamModal.

Bảng danh sách nhóm (CRUD):

Hiển thị tên nhóm, mô tả, số lượng thành viên, người tạo.

Hiển thị dự án liên quan: Nếu nhóm được liên kết với một dự án (qua team\_id trong PROJECTS), hiển thị tên dự án.

Hành động: Nút "Xem chi tiết", "Chỉnh sửa" (cho Admin/Editor), "Xóa" (cho Admin).

Modal "Tạo Nhóm Mới" (CreateTeamModal):

Bước 1: Thông tin nhóm:

Tên nhóm (input text, bắt buộc).

Mô tả (textarea, tùy chọn).

Bước 2: Thêm Thành viên vào Nhóm:

Lấy data từ danh sách nhân viên cá nhân: Khi mở modal, hiển thị danh sách nhân viên cá nhân (từ PersonalMemberList) để người dùng dễ dàng chọn.

Tick chọn để thêm nhiều: Cho phép chọn nhiều thành viên bằng cách tích vào checkbox.

Ô tìm kiếm thành viên: Bổ sung ô tìm kiếm (dùng searchUsersToAdd API) để tìm kiếm các user không có trong PersonalMemberList hoặc để lọc nhanh.

Gán vai trò: Mỗi thành viên được thêm vào danh sách tạm thời trong modal có một dropdown nhỏ để chọn vai trò (Editor, Member). (Vai trò Admin mặc định là người tạo nhóm).

Hiển thị danh sách thành viên đã chọn: Bao gồm tên, email, vai trò và nút "Xóa" khỏi danh sách tạm thời.

Lưu ý: Loại trừ người dùng hiện đang đăng nhập (người tạo nhóm) khỏi danh sách tìm kiếm/chọn.

Bước 3: Liên kết với Dự án (Tùy chọn):

Dropdown "Chọn Dự án hiện có": Liệt kê các dự án mà user hiện tại có quyền truy cập.

Nút "Tạo Dự án Nhanh":

Khi click, form 6 trường của dự án (tên dự án, mô tả, ngày bắt đầu, ngày kết thúc, trạng thái, ưu tiên, phân loại dự án) sẽ hiển thị ngay trong modal này.

Sau khi tạo thành công, dự án mới này sẽ tự động được chọn trong dropdown "Chọn Dự án hiện có" và team\_id của nhóm đang tạo sẽ được gán vào dự án này.

IV. Logic Backend (API):

POST /api/teams (Tạo nhóm mới):

Input: team\_name, description, members (mảng các { user\_id, role }), project\_id (tùy chọn), new\_project\_data (tùy chọn).

Logic:

Tạo TEAMS entry, created\_by là req.user.id.

Tự động thêm người tạo nhóm: Thêm req.user.id vào TEAM\_MEMBERS với role: 'Admin'.

Thêm các members được cung cấp vào TEAM\_MEMBERS (nếu chưa tồn tại, với vai trò được chọn).

Nếu có new\_project\_data: Gọi logic tạo dự án để tạo dự án mới. Gán team\_id của nhóm vừa tạo vào project\_id của dự án mới.

Nếu project\_id được cung cấp (chọn dự án có sẵn): Cập nhật trường team\_id trong PROJECTS cho dự án đó.

GET /api/teams (Lấy danh sách nhóm):

Input: page, limit, search, status (active/deleted).

Logic: Tìm kiếm TEAMS mà req.user.id là created\_by hoặc là user\_id trong TEAM\_MEMBERS. Dùng lookup để populate TEAM\_MEMBERS (bao gồm thông tin USERS của member) và PROJECTS liên quan (nếu có team\_id trong PROJECTS).

GET /api/teams/:id (Lấy chi tiết nhóm):

Logic: Tìm nhóm theo \_id, kiểm tra quyền của req.user.id. Populate TEAM\_MEMBERS (bao gồm thông tin USERS của member) và thông tin dự án nếu có liên kết.

PUT /api/teams/:id (Cập nhật thông tin nhóm):

Input: team\_name, description.

Logic: Chỉ Admin nhóm có quyền.

DELETE /api/teams/:id (Soft delete nhóm):

Logic: Chỉ Admin nhóm có quyền. Cập nhật trường is\_deleted: true cho nhóm.

POST /api/teams/:teamId/members (Thêm thành viên vào nhóm):

Input: user\_id, role.

Logic: Chỉ Admin hoặc Editor nhóm có quyền. Kiểm tra xem user đã tồn tại trong nhóm chưa.

PUT /api/teams/:teamId/members/:memberId (Cập nhật vai trò thành viên):

Input: role.

Logic: Chỉ Admin hoặc Editor nhóm có quyền. Editor không thể sửa vai trò Admin. Admin không thể sửa vai trò của chính mình.

DELETE /api/teams/:teamId/members/:memberId (Xóa thành viên khỏi nhóm):

Logic: Chỉ Admin hoặc Editor nhóm có quyền. Editor không thể xóa Admin hoặc Editor khác. Admin không thể tự xóa mình. Xóa entry khỏi TEAM\_MEMBERS hoặc set is\_active: false.

V. Mối quan hệ "PersonalMemberList" và "Nhóm":

PersonalMemberList là danh sách các nhân viên mà một owner\_user\_id cá nhân đã thêm vào để dễ dàng quản lý và tái sử dụng.

Khi "Add member" vào nhóm, bạn sẽ sử dụng dữ liệu từ PersonalMemberList như một nguồn gợi ý/tiện ích đầu tiên.

Nếu người dùng muốn thêm một thành viên không có trong PersonalMemberList của họ, họ có thể sử dụng chức năng tìm kiếm (searchUsersToAdd) trực tiếp trong modal thêm thành viên của nhóm.

Tổng kết:

Kế hoạch này có vẻ rất toàn diện và hợp lý, bao gồm cả các cân nhắc về trải nghiệm người dùng, phân quyền, và cấu trúc dữ liệu. Việc tái sử dụng form tạo dự án sẽ giúp tiết kiệm thời gian phát triển. Hãy bắt tay vào triển khai từng phần, bắt đầu từ các API backend cơ bản cho nhóm, sau đó là giao diện người dùng và tích hợp các chức năng phức tạp hơn.

# Backend

### ****Lộ trình Phát triển Chức năng "Nhóm" (Teams) - Cập nhật****

### ****Phần 1: Backend (Ưu tiên làm trước)****

Mục tiêu chính của phần backend là tạo ra các API cho phép quản lý nhóm, thành viên nhóm và liên kết với dự án, đồng thời triển khai logic phân quyền.

**1. Tạo mới Model:** Team **(Nhóm)**

Đây là model cốt lõi cho chức năng nhóm của bạn.

* **File:** backend/models/team.model.js (hoặc đặt tên phù hợp với cấu trúc project của bạn)
* **Mục đích:** Lưu trữ thông tin về các nhóm làm việc.

JavaScript

const mongoose = require('mongoose');

const teamSchema = new mongoose.Schema({

team\_name: {

type: String,

required: true,

trim: true // Xóa khoảng trắng ở đầu/cuối

},

description: {

type: String,

default: ''

},

created\_by: {

type: mongoose.Schema.Types.ObjectId,

ref: 'User', // Tham chiếu đến model User (người tạo nhóm)

required: true

},

is\_deleted: {

type: Boolean,

default: false // Hỗ trợ soft delete nhóm (đề xuất của bạn)

},

}, {

timestamps: true // Tự động thêm createdAt và updatedAt

});

const Team = mongoose.model('Team', teamSchema);module.exports = Team;

**2. Tạo mới Model:** TeamMember **(Thành viên Nhóm)**

Model này sẽ quản lý mối quan hệ giữa người dùng và nhóm, cùng với vai trò của họ trong nhóm.

* **File:** backend/models/teamMember.model.js (hoặc đặt tên phù hợp)
* **Mục đích:** Quản lý thành viên trong nhóm và vai trò của họ.

JavaScript

const mongoose = require('mongoose');

const teamMemberSchema = new mongoose.Schema({

team\_id: {

type: mongoose.Schema.Types.ObjectId,

ref: 'Team', // Tham chiếu đến model Team

required: true

},

user\_id: {

type: mongoose.Schema.Types.ObjectId,

ref: 'User', // Tham chiếu đến model User

required: true

},

role: {

type: String,

enum: ['Admin', 'Editor', 'Member'], // Vai trò trong nhóm: Admin (duy nhất), Editor, Member

default: 'Member',

required: true // Vai trò luôn phải có

},

joined\_at: {

type: Date,

default: Date.now // Thời điểm thành viên gia nhập nhóm

},

is\_active: {

type: Boolean,

default: true // Trạng thái hoạt động của thành viên trong nhóm (có thể dùng cho soft delete thành viên)

},

approval\_status: { // Nếu bạn muốn có quy trình phê duyệt thành viên

type: Boolean,

default: true // Mặc định là true nếu không có quy trình duyệt thủ công

}

}, {

timestamps: true // Tự động thêm createdAt và updatedAt

});

// Đảm bảo mỗi user chỉ là thành viên một lần trong mỗi nhóm (cặp team\_id và user\_id là duy nhất)

teamMemberSchema.index({ team\_id: 1, user\_id: 1 }, { unique: true });

const TeamMember = mongoose.model('TeamMember', teamMemberSchema);module.exports = TeamMember;

**3. Cập nhật Model:** Project **(Dự án)**

Bạn đã có Model Project. Chúng ta cần thêm một trường để liên kết dự án với nhóm, hỗ trợ chức năng "tạo dự án nhanh" và hiển thị nhóm thuộc dự án nào.

* **File:** backend/models/project.model.js (hoặc file bạn đã tạo cho Project)
* **Mục đích:** Thêm liên kết với Team.

JavaScript

const mongoose = require('mongoose');

// Giả định đây là schema Project hiện tại của bạn, tôi sẽ thêm trường mớiconst projectSchema = new mongoose.Schema({

project\_name: {

type: String,

required: true

},

description: {

type: String

},

start\_date: {

type: Date

},

end\_date: {

type: Date

},

status: {

type: String,

enum: ['Active', 'Archived'],

default: 'Active'

},

priority: {

type: String,

enum: ['Low', 'Medium', 'High'],

default: 'Medium'

},

is\_deleted: {

type: Boolean,

default: false

},

project\_type: { // Loại dự án (ví dụ: IT, Sale, Marketing)

type: String

// Bạn có thể cân nhắc thêm enum hoặc ref đến một ProjectType model nếu cần

},

// member\_count: { type: Number, default: 0 }, // Nếu bạn muốn lưu số lượng thành viên trực tiếp ở đây

// ---- TRƯỜNG MỚI ĐỂ LIÊN KẾT VỚI NHÓM ----

team\_id: {

type: mongoose.Schema.Types.ObjectId,

ref: 'Team', // Tham chiếu đến model Team

default: null // Cho phép dự án không thuộc nhóm nào

}

// ---- HẾT TRƯỜNG MỚI ----

// ... các trường Project khác của bạn

}, {

timestamps: true // Tự động thêm createdAt và updatedAt

});

const Project = mongoose.model('Project', projectSchema);module.exports = Project;

**4. Model:** PersonalMemberList **(Danh sách nhân viên cá nhân)**

Bạn đã đề cập rằng bạn đã làm model này. Tôi sẽ không hướng dẫn lại việc tạo model này, nhưng bạn cần đảm bảo rằng nó có các trường như owner\_user\_id, member\_user\_id, is\_active, custom\_role, notes và có unique index trên (owner\_user\_id, member\_user\_id) để đảm bảo tính toàn vẹn dữ liệu. Model này sẽ là nguồn dữ liệu để bạn lấy danh sách thành viên khi thêm vào nhóm.

**Các bước tiếp theo cho Backend:**

Sau khi bạn đã tạo và cập nhật các Schema Model này, hãy đảm bảo bạn import chúng vào các Controller và Router files.

Tiếp theo là phần quan trọng nhất của Backend: **Xây dựng Logic APIs** để tương tác với các Model này.

**API Controllers (Ví dụ:** backend/controllers/team.controller.js **và** backend/routes/team.routes.js**)**

**1. API Tạo Nhóm Mới (**POST /api/teams**)**

* + **Logic:**
    - Nhận team\_name, description, members (mảng các { user\_id, role }), new\_project\_data (nếu có), project\_id (nếu có).
    - Tạo Team mới.
    - **Tự động tạo** TeamMember **cho người tạo nhóm (**req.user.id**) với** role: 'Admin'.
    - Lặp qua mảng members: Tạo TeamMember cho từng người, với role được chọn. Bỏ qua nếu user đó đã là thành viên (do unique index).
    - **Xử lý "Tạo dự án nhanh":**
      * Nếu new\_project\_data tồn tại: Gọi một hàm hoặc service để tạo Project mới với 6 trường đã cho. Sau khi tạo, lấy \_id của Project đó.
      * **Gán** team\_id**:** Gán \_id của Team vừa tạo vào trường team\_id của Project mới này.
    - **Xử lý "Chọn dự án hiện có":**
      * Nếu project\_id được cung cấp: Cập nhật trường team\_id của Project đó với \_id của Team vừa tạo.
    - Trả về thông tin nhóm đã tạo (có thể populate các thành viên để frontend hiển thị).

**2. API Lấy Danh sách Nhóm (**GET /api/teams**)**

* + **Input:** page, limit, search, status (active/deleted), sortBy, sortOrder.
  + **Logic:**
    - Tìm kiếm Team mà req.user.id là created\_by HOẶC req.user.id là user\_id trong TeamMember của nhóm đó.
    - Sử dụng $lookup để populate TeamMember (và từ đó populate User của từng thành viên).
    - Sử dụng $lookup để populate Project (liên kết qua team\_id của Project khớp với \_id của Team).
    - Thực hiện tìm kiếm theo team\_name, description và phân trang, sắp xếp.
    - Trả về danh sách nhóm với thông tin phân trang.

**3. API Lấy Chi tiết Nhóm (**GET /api/teams/:id**)**

* + **Input:** id (Team ID).
  + **Logic:**
    - Tìm Team theo \_id.
    - **Kiểm tra quyền:** req.user.id phải là thành viên của nhóm.
    - Populate TeamMember (và User của thành viên) và Project liên quan.
    - Trả về thông tin chi tiết nhóm.

**4. API Cập nhật Thông tin Nhóm (**PUT /api/teams/:id**)**

* + **Input:** id (Team ID), team\_name, description.
  + **Logic:**
    - **Kiểm tra quyền:** User hiện tại phải có vai trò Admin trong nhóm.
    - Cập nhật team\_name và description.
    - Trả về nhóm đã cập nhật.

**5. API Xóa Nhóm (Soft Delete) (**DELETE /api/teams/:id**)**

* + **Input:** id (Team ID).
  + **Logic:**
    - **Kiểm tra quyền:** User hiện tại phải có vai trò Admin trong nhóm.
    - Cập nhật is\_deleted: true cho Team.
    - Cũng nên cập nhật is\_active: false cho tất cả TeamMember thuộc nhóm này.
    - Trả về thông báo thành công.

**API Controllers cho Thành viên Nhóm (Có thể nằm trong cùng** team.controller.js **hoặc file riêng** teamMember.controller.js**)**

**6. API Thêm Thành viên vào Nhóm (**POST /api/teams/:teamId/members**)**

* + **Input:** teamId, user\_id, role.
  + **Logic:**
    - **Kiểm tra quyền:** User hiện tại phải là Admin hoặc Editor của nhóm.
    - Kiểm tra user\_id có tồn tại và chưa phải là thành viên của nhóm.
    - Tạo TeamMember mới.
    - Trả về thành viên đã thêm.

**7. API Cập nhật Vai trò Thành viên (**PUT /api/teams/:teamId/members/:memberId**)**

* + **Input:** teamId, memberId (ID của TeamMember entry), role.
  + **Logic:**
    - **Kiểm tra quyền:** User hiện tại phải là Admin hoặc Editor của nhóm.
    - **Quy tắc phân quyền bổ sung:**
      * Nếu người yêu cầu là Editor: Không được phép cập nhật role của Admin.
      * Nếu người yêu cầu là Admin: Không được phép tự cập nhật role của chính mình.
    - Cập nhật role cho TeamMember.
    - Trả về TeamMember đã cập nhật.

**8. API Xóa Thành viên khỏi Nhóm (**DELETE /api/teams/:teamId/members/:memberId**)**

* + **Input:** teamId, memberId (ID của TeamMember entry).
  + **Logic:**
    - **Kiểm tra quyền:** User hiện tại phải là Admin hoặc Editor của nhóm.
    - **Quy tắc phân quyền bổ sung:**
      * Nếu người yêu cầu là Editor: Không được phép xóa Admin hoặc Editor khác.
      * Nếu người yêu cầu là Admin: Không được phép tự xóa mình khỏi nhóm.
    - Thực hiện soft delete (set is\_active: false) hoặc xóa cứng TeamMember entry.
    - Trả về thông báo thành công.

# Frontend

Tuyệt vời! Sau khi hoàn thành phần Backend, chúng ta sẽ chuyển sang Frontend để xây dựng giao diện người dùng cho chức năng "Nhóm" với trọng tâm là UI/UX mượt mà và trực quan.

Dưới đây là mô tả chi tiết về UI/UX cho các phần của chức năng "Nhóm", dựa trên các quyết định và ý tưởng của bạn:

### ****Phần 2: Frontend - Thiết kế UI/UX cho Chức năng "Nhóm" (Teams)****

**Mục tiêu chung về UI/UX:**

* **Tính nhất quán:** Duy trì phong cách thiết kế hiện có của UniPlan (Header, Sidebar, Breadcrumb, các widget).
* **Trực quan:** Dễ hiểu, dễ sử dụng cho người dùng.
* **Hiệu quả:** Giảm số bước, tối ưu hóa quy trình.
* **Phản hồi:** Cung cấp thông báo rõ ràng về các hành động của người dùng.
* **Responsive:** Đảm bảo hiển thị tốt trên các kích thước màn hình khác nhau.

**1. Cập nhật Thanh điều hướng (Header & Sidebar) phần này làm rồi chỉ nói cho biết quy trình thôi ko cần đụng tới(**file team trong thư mục teams**)**

* **Header (Dropdown "Tạo mới")**: Thêm tùy chọn "Tạo nhóm mới". Khi click, sẽ mở CreateTeamModal.
* **Sidebar**: Thêm một mục "Nhóm" (hoặc "Teams") với biểu tượng phù hợp (ví dụ: users hoặc briefcase). Khi click vào đây, sẽ dẫn đến trang /teams (Trang Quản lý Nhóm chính).

**2. Trang Quản lý Nhóm (**/teams**)**

Đây là trang chính hiển thị danh sách các nhóm mà người dùng thuộc về hoặc đã tạo.

* **Breadcrumb**: Sẽ hiển thị Dashboard > Nhóm. (phần này đã làm rồi chỉ nói cho biết ko cần đụng tới làm theo file team trong thư mục teams)
* **Tiêu đề trang**: "Danh sách Nhóm của bạn".
* **Mô tả ngắn gọn**: "Quản lý các nhóm làm việc để tổ chức dự án và thành viên hiệu quả hơn."
* **Nút "Tạo Nhóm Mới"**: Nút lớn, nổi bật (ví dụ: màu xanh lam) ở góc trên bên phải, giống nút "Thêm Nhân viên", khi click sẽ mở CreateTeamModal.
* **Thanh tìm kiếm**: Một ô input "Tìm kiếm nhóm..." cho phép người dùng tìm kiếm theo tên nhóm hoặc mô tả.
* **Bộ lọc/Sắp xếp**: Các tùy chọn để lọc nhóm theo trạng thái (ví dụ: "Đang hoạt động", "Đã xóa") và sắp xếp theo "Tên", "Ngày tạo", "Số lượng thành viên".
* **Bảng danh sách nhóm**:
  + **Cấu trúc bảng**: Các cột rõ ràng như "Tên Nhóm", "Mô tả", "Số lượng Thành viên", "Người tạo", "Trạng thái", "Dự án liên quan", "Hành động".
  + **Hiển thị dữ liệu**:
    - **Tên Nhóm**: Dạng liên kết, khi click vào sẽ dẫn đến trang chi tiết nhóm (/teams/:teamId).
    - **Số lượng Thành viên**: Con số hiển thị số lượng thành viên đang hoạt động trong nhóm.
    - **Người tạo**: Tên của user đã tạo nhóm.
    - **Trạng thái**: "Đang hoạt động" hoặc "Đã Hoàn Thành" (nếu bạn soft delete).
    - **Dự án liên quan**: Hiển thị tên dự án nếu nhóm đó được liên kết với một dự án (dựa trên trường team\_id trong PROJECTS). Nếu không có, có thể hiển thị "Không có" hoặc "--". Tên dự án cũng có thể là liên kết tới trang chi tiết dự án.
    - **Hành động**: Các biểu tượng bút chì (Chỉnh sửa), thùng rác (Xóa).
      * **Hiển thị có điều kiện**:
        + Chỉnh sửa: Chỉ hiển thị cho Admin và Editor của nhóm.
        + Xóa: Chỉ hiển thị cho Admin của nhóm.
  + **Trạng thái rỗng**: Nếu không có nhóm nào, hiển thị EmptyState tương tự như danh sách nhân viên, với thông báo "Bạn chưa có nhóm nào. Hãy tạo nhóm đầu tiên!" và một nút "Tạo Nhóm Mới".
* **Phân trang**: Tương tự như danh sách nhân viên, hiển thị điều khiển phân trang ở cuối bảng nếu có nhiều hơn một trang dữ liệu.

**3. Modal "Tạo Nhóm Mới" (**CreateTeamModal**)**

Modal này sẽ là trung tâm của trải nghiệm tạo nhóm, cho phép người dùng tạo nhóm và liên kết dự án một cách liền mạch.

**Thiết kế chung**: Modal bật lên, phủ mờ phần còn lại của giao diện.

**Tiêu đề**: "Tạo Nhóm Mới".

**Các phần chính trong Modal**:

**a. Thông tin Nhóm**:

* + - **Tên Nhóm**: input text, placeholder "Nhập tên nhóm của bạn".
    - **Mô tả Nhóm**: textarea, placeholder "Mô tả ngắn gọn về mục tiêu của nhóm".
    - **Gợi ý UI/UX**:
      * Sử dụng nhãn rõ ràng, placeholder hữu ích.
      * Hiển thị thông báo lỗi ngay lập tức nếu tên nhóm bị bỏ trống.

**b. Thêm Thành viên vào Nhóm**:

* + - **Tiêu đề**: "Thêm Thành viên" (hoặc "Thêm thành viên vào nhóm").
    - **Phần "Chọn từ danh sách cá nhân"**:
      * Một bảng nhỏ hoặc danh sách cuộn hiển thị PersonalMembersList của người dùng hiện tại.
      * **Checkbox**: Mỗi thành viên có một checkbox ở đầu hàng, cho phép người dùng tích chọn nhiều người cùng lúc.
      * **Tìm kiếm/Lọc trong danh sách cá nhân**: Một ô tìm kiếm nhỏ để lọc nhanh danh sách nhân viên cá nhân.
      * **Cột hiển thị**: "Tên", "Email", "Vai trò tùy chỉnh" (từ PersonalMemberList).
    - **Phần "Hoặc tìm kiếm người dùng khác"**:
      * Một ô input "Tìm kiếm người dùng..." (search bar) cho phép người dùng nhập tên hoặc email.
      * Khi người dùng gõ, gọi API searchUsersToAdd (frontend sẽ tự động debounce input để tránh gọi API liên tục).
      * **Danh sách kết quả tìm kiếm**: Hiển thị danh sách thả xuống (hoặc dưới ô tìm kiếm) các user phù hợp (tên, email) mà CHƯA PHẢI là thành viên của nhóm đang được tạo và KHÔNG PHẢI là user hiện tại.
      * Khi click vào một kết quả, user đó được thêm vào danh sách thành viên tạm thời (phía dưới).
    - **Danh sách Thành viên Đã Chọn (Tạm thời)**:
      * Hiển thị tên, email của các thành viên đã được chọn (từ PersonalMemberList hoặc qua tìm kiếm).
      * **Dropdown "Vai trò"**: Bên cạnh mỗi thành viên, có một dropdown nhỏ với các lựa chọn: Editor, Member. (Lưu ý: Admin là mặc định cho người tạo nhóm và không có trong dropdown này).
      * **Nút "Xóa"**: Biểu tượng x hoặc thùng rác bên cạnh mỗi thành viên để xóa họ khỏi danh sách tạm thời trước khi tạo nhóm.
    - **Gợi ý UI/UX**:
      * Hiển thị số lượng thành viên đã chọn.
      * Khu vực cuộn nếu danh sách thành viên tạm thời quá dài.

**c. Liên kết với Dự án (Tùy chọn)**:

* + - **Dropdown "Chọn Dự án hiện có"**:
      * Hiển thị danh sách các dự án mà người dùng có quyền truy cập.
      * Giá trị mặc định là "Không liên kết dự án" hoặc "Chọn dự án...".
    - **Nút "Tạo Dự án Nhanh" ( hoặc tái dự dụng form ProjectForm.tsx)**:
      * Một nút nhỏ (ví dụ: "Tạo nhanh Project") bên cạnh hoặc dưới dropdown.
      * **Khi click**: Một form dự án 6 trường sẽ xuất hiện **ngay trong modal này**.
        + **Tên dự án** (input text, bắt buộc).
        + **Mô tả** (textarea, tùy chọn).
        + **Ngày bắt đầu & Ngày kết thúc** (date pickers).
        + **Trạng thái** (dropdown: Active, Archived).
        + **Độ ưu tiên** (dropdown: Low, Medium, High).
        + **Phân loại dự án** (dropdown: IT, Sale, Marketing, v.v. - có thể là một dropdown lấy data từ backend hoặc cho phép tạo mới nếu cần).
      * **Gợi ý UI/UX**:
        + Có thể dùng hiệu ứng chuyển động nhẹ (ví dụ: Framer Motion slide-down) khi form dự án hiện ra.
        + Nút "Tạo Project" bên trong form này.
        + Sau khi tạo thành công, form dự án biến mất, và dropdown "Chọn Dự án hiện có" sẽ tự động chọn dự án vừa được tạo.

**d. Nút Hành động của Modal**:

* + - Nút "Hủy" (Cancel): Đóng modal.
    - Nút "Tạo Nhóm" (Create Team): Kích hoạt việc gửi dữ liệu đến API POST /api/teams.
    - **Trạng thái tải**: Nút "Tạo Nhóm" có thể hiển thị trạng thái loading (spinner) khi đang gửi dữ liệu.

**4. Modal "Chỉnh sửa Nhóm" (**EditTeamModal**)**

* **Thiết kế**: Tương tự như CreateTeamModal.
* **Dữ liệu**: Khi mở modal, tất cả các trường (tên nhóm, mô tả, thành viên, dự án liên kết) sẽ được pre-fill với dữ liệu hiện có của nhóm.
* **Chức năng**:
  + Chỉnh sửa team\_name, description.
  + Thêm/xóa thành viên khỏi danh sách (giao diện tương tự như CreateTeamModal nhưng áp dụng cho nhóm hiện tại).
  + Cập nhật vai trò của thành viên hiện có (áp dụng các quy tắc phân quyền: Editor không sửa Admin, Admin không tự sửa mình).
* **Nút Hành động**: "Hủy", "Cập nhật Nhóm".

**5. Modal "Xác nhận Xóa Nhóm" (**ConfirmDeleteTeamModal**)**

* **Thiết kế**: Modal nhỏ, đơn giản.
* **Nội dung**: "Bạn có chắc chắn muốn xóa nhóm [Tên Nhóm] không? Hành động này sẽ ẩn nhóm khỏi danh sách của bạn."
* **Nút Hành động**: "Hủy", "Xóa".

**6. Thông báo và Phản hồi**

* Sử dụng các toast/snackbar hoặc alert nổi bật để thông báo cho người dùng về kết quả của các hành động (thành công, thất bại, lỗi quyền, lỗi validation). Ví dụ:
  + "Đã tạo nhóm thành công!"
  + "Lỗi: Tên nhóm không được để trống."
  + "Bạn không có quyền thực hiện hành động này."
  + "Thành viên đã tồn tại trong nhóm."

Đây là một mô tả chi tiết về UI/UX để bạn có thể hình dung và bắt đầu phát triển frontend cho chức năng "Nhóm". Chúc bạn thành công!