**Demo project với mô hình phân quyền RBAC**

DB cho RBAC với cấu trúc bao gồm các colection sau:

1. User : lưu trữ thông tin tài khoản người dùng
2. Role : lưu trữ các quyền
3. UserRole : lưu thông tin User tương ứng với những Role nào
4. Department : lưu trữ các phòng và ban
5. Privilege : lưu trữ phân quyền tương ứng với từng loại resoucre ( có thể là Link, TaskTemplate, ComponentUI, … )
6. TaskTemplate : lưu thông tin mẫu công việc
7. Link : lưu các link truy cập của hệ thống để cấp cho các role tương ứng.
8. Action : lưu các action tương ứng với resource của hệ thống.

Mô tả chi tiết về các colection:

1. User

* \_id : lưu id của user
* name : lưu tên của user
* email : lưu email
* password : lưu mật khẩu

1. Role

* \_id : lưu id của role
* name : tên của role
* abstract: lưu các id của các role mà role hiện tại sẽ có tất cả các quyền của các role đó. Ví dụ : Dean sẽ abstract ViceDean và Employee -> Dean sẽ có tất cả các quyền mà ViceDean và Employee có thể làm.

1. UserRole

* \_id : lưu id của cặp user\_role tương ứng
* id\_user : mảng lưu id của các user được cấp role tương ứng(phía dưới)
* id\_role : lưu id của role tương ứng với các user

1. Department

* \_id : lưu id của phòng, ban
* name : lưu tên của phòng, ban
* dean : lưu id của role Dean tương ứng với department
* vice\_dean : lưu id của role ViceDean tương ứng với department
* employee : lưu id của role Employee tương ứng với department

1. Privileges

* \_id : lưu id của privilege
* resource : lưu id loại resource ( tài nguyên : link, task template, component UI, … )
* resource\_type : lưu tên loại resource ( tương ứng với tên của Colection resource : link, task template, componentUI, …)
* role : một mảng lưu id của các role tương ứng với phân quyền
* action : lưu id của hành động tương ứng của role với resource

1. TaskTemplate

* id : lưu id của mẫu công việc
* name : tên của mẫu công việc
* creator : lưu id của user đã tạo
* description : mô tả của mẫu công việc
* status : trạng thái của mẫu công việc ( true: mở, false: khóa )

1. Link

* \_id : lưu id của link
* url : lưu url ( ví dụ : /home, /task-template, … )

1. Action

* \_id : lưu id của action
* name : lưu tên của action
* see : có thể xem
* open : có thể mở
* edit : có thể sửa
* delete : có thể xóa
* enable : có thể kích hoạt mở resource
* disable : có thể tạm đóng resource

**Về phân quyền RBAC trong hệ thống**

Sẽ có 4 super role được xậy dựng trong hệ thống là : Admin, Dean, ViceDean và Employee để thực hiện phân quyền về mặt giao diện trên hệ thống.

Ví dụ :

* Admin sẽ có quyền truy cập và thực hiện các chức năng của trang admin.
* Tương tự với Dean,ViceDean, Employee sẽ có quyền truy cập và thực hiện các chức năng của Dean ( trưởng phòng ) ,ViceDean( phó phòng ) và Employee ( nhân viên ).

Khi tạo một tài khoản mới thì tài khoản đó sẽ được gán 1 trong các super role trên ( có thể cộng với các role thường khác tương với các department mà tài khoản đó thuộc về ) .

Ví dụ : tạo một tài khoản là Nguyễn Văn A sẽ có một super role là Dean : để thực hiện các chức năng của một Dean ( Trưởng Phòng ) mà một hoặc nhiều role thường ( Dean of “Department name 1”, ViceDean of “Department name 2, vv… ) có nhiệm vụ quản lý các tài nguyên của department đó.

\*\*\*\*\*

1. Mô tả về cấu trúc backend( bên server )

**Cấu trúc thư mục bên server:**

* controller: lưu trữ các file controller có nhiệm vụ điều hướng từ các route đến các service tương ứng.
* logs: chứa file cấu hình cho việc ghi logger lịch sử hoạt động trên server.
* middleware: chứa các lớp trung gian có nhiệm vụ như : xác thực người dùng trước khi có quyền truy cập vào service của server, kiểm tra quyền của người dùng, …
* models: chứa các file model : cấu trúc của các collection trong database - định dạng lưu dữ liệu trên server.
* routes: chứa các file route
* seed: chứa các file seed - có nhiệm vụ tạo một cơ sở dữ liệu mẫu cho việc test, tạo dữ liệu gốc cho server.
* services: chứa các file services tương ứng với controller đảm nhiệm các dịch vụ , thực hiện các yêu cầu mà người dùng gửi đến cho server.
* validation: chứa các file có nhiệm vụ validate dữ liệu mà người dùng gửi đến - dữ liệu hợp lệ thì thực hiện service - dữ liệu không hợp lệ thì thông báo trả về yêu cầu người dùng nhập đúng theo định dạng.
* .env : chứa cấu hình cho server, các secret cho việc tạo token, tên của cơ sở dữ liệu,...
* package.json: lưu thông tin các package được cài đặt trong project.
* server.js : file thực thi khởi chạy cho server.

**Khởi động server**: npm start

**Cách viết một API cho bên server.**

* Khởi tạo route cho chức năng cần xây dựng trong thư mục routes. Tùy với route nào cần bảo mật thì sẽ add thêm các middleware vào đó.
* Định nghĩa API cho chức năng ( ví dụ : /login cho chức năng đăng nhập) trong thư mục routes.
* Chuyển đến thư mục controller định nghĩa một controller để thực hiện chuyển hướng đến service đảm nhiệm việc thực hiện chức năng( ví dụ : login ) mà người dùng gọi đến API tương ứng với route đã tạo.
* Chuyển đến thư mục service và viết API cho chức năng login nhận dữ liệu gửi đến từ người dùng và trả về giá trị . Trong các API có thể lồng thêm các logger để ghi lại lịch sử hoạt động file các file log.

1. Mô tả về cấu trúc frontend( bên client)