Assignment – MERN Stack

Instructions

In this assignment you will extend to verify an Admin and grant appropriate privileges to an Admin. In addition, you will allow only a member to update his/her information. Neither another member, nor an Admin can edit this information.

- You will now create the members Schema and model to support the document:

```
{
        "email": "admin@myteam.com",
        "password": "xxxxxxxxxxxxxxxxxxxxxxxxx",
        “name”: “Do Nam Trung”,
        “YOB”: 1990,
        “gender”: true
        “isAdmin”: true
}
```

Password must be hashed by using bcrypt module.

- Your database contains some collections, the schema as follow (Assignment 1):

```
const brandSchema = new Schema({ brandName: String},{ timestamps: true, });
const perfumechema = new Schema({
    perfumeName:{ type: String, require: true},
    uri:{ type: String, require: true},
    price: {type: Number, require: true},
    concentration:{type: String, require: true}, // nồng độ của nước hoa: Extrait
, EDP, EDT,…
    description:{type: String, require: true},
```

```
    ingredients:{ type: String, require: true},
    volume:{type: Number, require: true},
    targetAudience:{ type: String, require: true},// male, femail, unisex
    comments: [commentSchema]
   brand:{type: mongoose.Schema.Types.ObjectId, ref: "Brands", require: true},
},{ timestamps: true, });
commentSchema = new Schema({
    rating:{ type: Number,  min: 1,  max:3,  require: true},
    content: {type: String, require: true},
    author:{  type: mongoose.Schema.Types.ObjectId, ref: "Members", require: true }
},{timestamps: true}
)
const memberSchema = new Schema({
membername:{type: String, require: true}, password:{type: String, require: true}, isAdmin:{type: Boolean, default: false}},{ timestamps: true, });
```

Assignment Overview

At the end of this assignment, you would have completed the following:

- Implement the login action, using OAuth2 is a plus.
- Using Mongoose population to populate information into the perfume document from the referenced brand document.
- Check if a verified ordinary member also has Admin privileges.
- Allow anyone to perform GET operations on public routes.

- Allow only an Admin to perform GET, POST, PUT and DELETE operations in private routes.
- Allow an Admin to be able to GET all the registered members' information from the database.
- Allow a member to edit his/her information. They should be restricted from performing such operations only on his/her own account. No member or even the Admin can edit or delete the information by other members.
- Members can send their comments to perfumes. Only one comment on a perfume.

Assignment Requirements

This assignment is divided into four tasks as detailed below; all tasks should build with their UI:

Task 1 - Assignment 2:

In this task you will implement the public routes which all the users can access and som member pages, including:

- The index route that shows all of perfumes, the data will include name, image, targetAudience and brandName.
- The detailed route will display all the perfume's information.
- Members can search by perfume name.
- Members can filter by brand name
- Member can register an account, the default role is not Admin.
- Members can login by his/her account after successful registration.
- A member can edit his/her information after logging in successfully.
- A member can change his/her password.
- A member can manage his/her feedback and rating a perfume once.
- Outstanding design for the Extrait value (concentration property) of the perfume.

Task 2 - Assignment 3:

In this task you will update all the routes to ensure that only the Admin can perform GET, POST, PUT and DELETE operations. Update the code for all the routers to support this. These operations should be supported for the following endpoints:

- GET, POST, PUT and DELETE operations on /brands and /brands/:brandId
- GET, POST, PUT and DELETE operations on /perfumes and /perfumes/:perfumeId

Task 3 - Assignment 4:

In this task you will implement the feedback feature that includes the posting comment and rating function. Only members can feddback. Each member can only feedback one perfume once.

Task 4 - Assignment 4:

In this task you will now activate the /collectors endpoint. When an Admin sends a GET request to /collectors you will return the list of all the members. Ordinary members are forbidden from performing this operation.

Essential UI/UX elements that cannot be ignored! ☠☠☠