FPT UNIVERSITY

Assignment 3 - User Authentication

Instructions

In this assignment, you will continue exploring user authentication. We have already set up the REST API server to validate an ordinary user. Now, you will extend this functionality to verify an Admin and grant appropriate privileges to an Admin. Additionally, you will allow only an author to update and delete their submitted questions. Neither another user nor an Admin can edit these questions.

Assignment Overview

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

- Check if a verified ordinary user also has Admin privileges.
- Allow any one to perform GET operations
- Allow only an Admin to perform POST, PUT and DELETE operations
- Allow an Admin to be able to GET all the registered users' information from the database
- Allow an author to submit questions (already completed), update and delete a submitted question. The author should be restricted to perform such operations only on his/her own questions. No user or even the Admin can edit or delete the questions submitted by other users.

Assignment Requirements

This assignment is divided into three tasks as detailed below:

Task 1

Firstly, you should add author in Question model. Author will reference ObjectId from User

Your models will change:

User: Represents a single user that can use the quiz app.

- username field is defined with the String type and a default value of an empty string "".
- The admin field is defined with the Boolean type and a default value of false.

Question: Represents a single question in the quiz.

- Object_id: Unique identifier for the question.
- text: The text of the question.
- Author: ObjectId is referenced from User
- options: An array of answer options for the question.
- keywords: An array of keywords for the question.
- correctAnswerIndex: Index of the correct answer within the options array.

In this task you will implement a new function named ***verifyAdmin() and vertifyAuthor()*** in ***authenticate.js*** file.

The ***verifyAdmin()*** function will check an ordinary user to see if s/he has Admin privileges. In order to perform this check, note that all users have an additional field stored in their records named ***admin***, that is a boolean flag, set to ***false*** by default. Furthermore, when the user's token is checked in ***verifyUser()*** function, it will load a new property named ***user*** to the ***request*** object. This will be available to you if the ***verifyAdmin()*** follows ***verifyUser()*** in the middleware order in Express. From this ***req*** object, you can obtain the admin flag of the user's information by using the following expression:

```
req.user.admin
```

You can use this to decide if the user is an administrator. The *verifyAdmin()* function will call *next();* if the user is an Admin, otherwise it will return next(err); If an ordinary user performs this operation, you should return an error by calling next(err) with the status of 403, and a message "You are not authorized to perform this operation!".

The "verifyAuthor()" function will check an ordinary user to see if they are the author of a question. To perform this check, note that all users are checked in the "verifyUser()" function, which loads a new property named "user" to the request object. After that, retrieve the ObjectId of the question's author and compare it with the ObjectId of the user. If they match, the user can update the question; otherwise, the user will receive the error message "You are not the author of this question".

Task 2

In this task you will update all the routes in the REST API to ensure that only the Admins can perform POST, PUT and DELETE operations for Quiz. Update the code for all the routers to support this. These operations should be supported for the following end points:

- POST, PUT and DELETE operations on /quizzes and /quizzes/:quizId

Task 3

In this task you will now activate the /users REST API end point. When an Admin sends a GET request to http://localhost:3000/users you will return the details of all the users. Ordinary users are forbidden from performing this operation.

Task 4

Only author can update and delete his/her submitted questions. Update the code for all the routers to support this. These operations should be supported for the following end points:

- POST, PUT and DELETE operations on /questions and / questions /: questionId

Review criteria

Your assignment will be graded based on the following criteria:

Task 1

- You have implemented the *verifyAdmin()* function in *authenticate.js*.
- The *verifyAdmin()* function will allow you to proceed forward along the normal path of middleware execution if you are an Admin
- The *verifyAdmin()* function will prevent you from proceeding further if you do not have Admin privileges, and will send an error message to you in the reply.

Task 2

- Any one is restricted to perform only the GET operation on the resources/REST API end points.
- An Admin (who must be first checked to make sure is an ordinary user), can perform the GET, PUT, POST and DELETE operations on any of the resources/ REST API end points.

Task 3

- A GET operation on http://localhost:3000/users by an Admin will return the details of the registered users
- An ordinary user (without Admin privileges) cannot perform the GET operation on http://localhost:3000/users.

Task 4

- Only author is allowed to update and delete his/her own questions.
- Any user or an Admin cannot update or delete the question posted by other users.