Lab Assignment #3 – Designing and implementing JWT Based Authentication

Due Date: Please refer to ecentennial course shell (Late submissions will have penalties)

Marks: 12 Marks

Purpose:

The purpose of this assignment is to develop a JWT Based Authentication API

References: In class code uploaded to Week 6

Be sure to read the following general instructions carefully:
- This assignment must be completed individually by all the students.
- See the naming and **submission rules** at the end of this document

Instructions:

1. Create an API which is capable of registering and logging in users. You must use mongoose and MongoDB cloud.
2. User must have following properties – userId, roleId, email, password. There are two kind of users – admin and student. Create a mongoose schema. (1 mark)
3. Registering the user must be handled by '/signup' end point. The client should pass email, password & role while sending the POST request. Add validation for email, role and password. Role must be either 'Admin' or 'Student'. Password must be 6 characters minimum. Use bcrypt to salt and hash the password before storing it in the database. Never store passwords as plain text in the database. Once signup is successful send a 201 Success message with newly created user data as JSON (avoid sending password data). If validations fail during signup send relevant error messages as JSON and status codes. (3 marks)
4. User login must be handled by '/login' end point. When client sends a POST request to this end point, they will send email and password in the request body. Once your controller receives this request, you must check if the user exists in our database and if user is not found send 404 status and JSON with error message. If the user is found, compare the password provided by the client in this request, to the hashed password in the database (Tip: Use bcrypt). If the user is authenticated create JWT token, payload must include userId, roleId and IAT. Send the token as a cookie to the client.(3 marks)
5. Create a route called '/adminPortal' which returns a hardcoded JSON of your choice – only admin can see this page. When client sends a request for this route they must send the valid JWT token in HTTP headers. (2 marks)
6. Create a route called '/studentData' which returns a hardcoded JSON of your choice – Both admin & students can see this page. When client sends a request for this route they must send the valid JWT token in HTTP headers. (2 marks)
7. For Steps 5 and 6, send relevant error messages as JSON to Users when authentication or authorization fails.(1 mark).

Submission rules:

1. You must name your project as: YourFullName_COMP308AssignmentNumber

2. Remove the node_modules folder before zipping the project. (DO NOT use RAR or other types of archives)

3. Demonstration of Work (Mandatory):

Choose one of the following options.

Option 1: Explain the code and show the output inperson, during lab. (before due date) (OR)

Option 2: Submit a short screen recording – 2min to 3min, explaining the code. It is ok to keep it simple - Please don't spend much time on recording the video. You must upload the video to ecentennial if you choose this option. Youtube or any third party websites should not be used to share the video.

Think of these options as a practice session for your interview, where you are expected to give a code walkthrough

Penalty applicable if submissions rules are not followed.