Authentication

In this assignment, you will validate your knowledge and implement authentication-related functionalities, including account registration and login, email verification, and authorization middleware.

Account Registration

- Create a service to register an account for each user type (`Coder` and `Manager`).
- Make sure to return proper responses when the user supplies an already existing email as we don't allow duplicate emails in the database.
- The account should be registered as `unverified`.
- The password should be encrypted. You can use bcrypt
- You should send a verification email to the user containing a token to verify it.

For email verification, you should generate a JWT token that encodes the id and the role of the user.

- Next, create a token-parameterized path (a route that accepts a token as a URL param) with that token. Something like `http://localhost:8080/verify?token=<Your Token>`
- Use nodemailer to send an email containing that token.
- Create a verification route that meets the verification path, you should verify the token, search for the user by the id encoded in the token and if everything is good, update the `is_verified` field to true.

- Make sure to return proper responses in case of success and errors (You can return HTML templates that contain the messages).

Account Login

- Create a service to log in each user type (`Coder` and `Manager`).

- Make sure to return proper responses when the user has not yet verified his email or gives the wrong credentials.

- If the user is verified and has supplied the correct credentials, you should create a JWT token that encodes two important information, the user id and his email.

Authorization Middleware

Next, you have to implement authentication and authorization middleware.

This middleware will be very important to guard the endpoints.

- Create an express middleware creator that takes the set of authorized roles as a parameter and based on the token present in the request, it `allows` or `denies` the request.

- The middleware should **inject** user info (id and email) extracted from the token into the request object.