Role-Based Access Control (RBAC) Assignment

In this assignment, you are required to implement a Role-Based Access Control (RBAC) system in a blog platform. The goal is to create a secure application where different roles have distinct permissions, such as admin and user. This assignment will evaluate your ability to handle authentication, authorization, and secure data access.

Objectives

1. Implement authentication and role-based authorization for users and admins.

2. Ensure secure access to different functionalities based on user roles.

3. Develop a full-stack solution integrating frontend and backend.

Requirements

1. **Backend**: Use Node.js and Express.js to handle authentication and role-based authorization. Implement the following:

   - Authentication using JWT.

   - Role assignment for users (e.g., admin, user).

   - Middleware to restrict access based on roles (e.g., only admins can create or delete blog posts).

2. **Frontend**: Create a React-based user interface with the following features:

   - Login and signup forms.

   - Pages for users to view blog posts.

   - An admin dashboard to manage blog posts (create, update, delete).

3. **Database**: Use MongoDB or PostgreSQL to store user roles, blog posts, and user information. The data model should include:

   - User details (name, email, role, hashed password).

   - Blog post details (title, content, author, timestamp).

4. **Optional**: Implement email verification for user signup.

Deliverables

1. Source code for the frontend and backend.

2. README file with instructions on how to run the application locally.

3. A brief document explaining the architecture and flow of the application.

Evaluation Criteria

1. Code Quality: Readability, maintainability, and adherence to coding standards.

2. Functionality: Implementation of role-based access control and required features.

3. Security: Proper handling of authentication, authorization, and sensitive data.

4. User Experience: Design and usability of the frontend interface.

5. Innovation: Bonus points for additional features such as email verification or real-time updates.

Submission Guidelines

1. Share the project code via a GitHub repository (ensure the repository is public or accessible).

2. Include a clear README file with instructions to run the project.

3. Submit the repository link via email or as per the instructions provided.

4. **Due Date:** [Insert Due Date Here]