Employee Information Portal

A CS814 Course Project Report

Submitted by:

Ajay Sherawat(202IS003)
Tridiv Kumar Rabha(202IS027)

Department of Computer Science and Engineering
National Institute of Technology Karnataka
P.O. Srinivasnagar, Surathkal, Mangalore-575025
Karnataka, India

January 2021

Table of Contents

1. Introduction

a. Use of Application

This is a Role-Based Access Control application using Nodejs, Express, Passport Js, etc. This application can be used for any organization which needs authentication and authorization.

The application is used to show details of members of an organization. The details can only be seen by a member of the organization and not disclosed to anyone outside of the organization. These details are maintained by an admin which is one of the members of the organization.

b. Types of users and their functionalities

There are two types of user in the application mentioned below -

- ➢ Admin - It is a member of the organization which has the role of maintenance of data of the members in the organization.
  It has the following functionalities:
  - Manage Users - The admin can manage all the details of a new member of the organization. Admin can view all the user's role from the database.

  - Update Role - The admin can update the role of a member of the organization to the Client or Admin. Giving an admin the role of Client will take away his admin rights and giving a Client the Admin role will give the user admin privileges.

- ➢ Client - It is a member of the organization who can view the list of all the members of the organization and his profile details. A Client has the functionality of viewing the details only.

2. Authorization

a. Need for RBAC based authorization

RBAC is used to predefine relationships which makes it easier to assign users to predefined roles. This is because the permissions assigned to roles tend to change relatively slower than user membership in roles. It is convenient to let administrators confer and revoke user membership in existing roles without letting administrators change roles or letting them assign new permission to the

existing roles. Without using RBAC it is difficult to determine which permissions are assigned to which users.
In the application, RBAC based authorization is needed to assign the roles to the users. These users have permissions associated with the roles and only the users with admin as the role can change the roles of the other users.

b. Components of RBAC in the application

The application consists of role-permission and user-role components of RBAC. The role-permission for the different roles is - admin role and client role. Anyone who registers for the first time will be assigned to the client role. Admin role has 3 permissions associated with it, namely, to Update Roles and View the profile of every user. The client role has only one permission associated with it, i.e, View profile Details.
The user-role component is incorporated as there all the users are assigned to either Client or Admin role and only Admin can change the roles of the users. Admin can not remove themselves from admin.

c. Components of Administrative Model

It is worth emphasizing that RBAC distinguishes roles and permissions from administrator roles and permissions, respectively, where the latter is used to manage the former. RBAC has many components. In the application, there is an admin role with administrative permissions assigned to it. The admin can update the roles of the existing users to the Client or admin.

3. Conclusion

The application has successfully utilized the components of the RBAC model to make it easier to assign the permissions to users by assigning them roles. These roles have permissions associated with them which can only be given to the user having the role to perform these tasks.
The application can be further improved by adding more functionality to it if the need arises. More roles can be added if required and permissions are assigned to these roles. RBAC model makes it very convenient to keep track of all the permissions assigned to a user by assigning these permissions to the roles.
Further work related to our project can be to give Add, Delete, and Update employee

information permissions to the admin to display the data of the employees on the front page of the app. Also, we can include administrative components to the app by giving the add, delete, and update roles in the app and assigning the permissions to these roles.

4. References

i. Ravi S. Sandhu, Edward J Coyne " Role-Based access control models".
ii. https://docs.npmjs.com/
iii. https://docs.mongodb.com/manual/
iv. http://www.passportjs.org/docs/
v. https://www.npmjs.com/package/ejs