Documentation RBAC (Role Based Access Control)

1. Introduction

Overview

This documentation outlines the implementation of a User Role and Permission System in a Laravel-based application. The system is designed to manage user access and permissions efficiently.

Purpose of the System

The primary purpose is to provide a robust framework for assigning roles and permissions to users, ensuring secure access control within the application.

2. Installation

Prerequisites

Before starting, ensure that you have a working Laravel environment with PHP, Composer, and a compatible database (e.g., MySQL) installed.

Installing Laravel

Follow Laravel's official documentation to install the framework using Composer.

Setting up the Database

Use Laravel's migration system to create the necessary tables for users, roles, and permissions.

3. User Authentication

Laravel's Built-in Authentication System

Utilize Laravel's built-in authentication system for user registration and login functionality.

User Registration Only Super Admin Create New User.

User Login

An user can login using his login email & password.

4. User Roles

Creating Admin, Manager and Employee any Dynamic Roles creating by Super Admin

Define the distinct roles all are dynamic within the system, such as Admin, Manager, and Employee.

5. Permissions

Defining Permissions for Each Role

Specify the actions and features associated with each role. For example, Super Admins can perform all actions, All other role have limited access.

Role-Based Access Control (RBAC)

Implement RBAC using Laravel's Gate or Policy system to enforce permissions follow bellow step

- In a AppServicesProvider.php file all request check to permissions by gate facads
- In a AuthserviceProverder.php file all model can acting an policy with Super Admin Gate define all access.

6. User Interface

Designing Admin Interface

Create a user-friendly admin interface where administrators can manage roles, permissions, and users.

Assigning Roles and Permissions

An Super admin interface create/update roles, permissions all access by simple UI.

CRUD Operations for Post Table

Implementation all role in Post CRUD interface for creating, reading, updating, and deleting.

7. Middleware

Implementing Middleware for Role-Based Access

Create middleware for Super Admin to restrict access to certain routes or actions based on the others user's role and permissions.

8. Database

Laravel Migration for Users, Roles, Post, Permissions with pivot table.

Provide the migration scripts for creating the necessary database tables.

9. Conclusion

Summary

RBAC the key points covered in the documentation full functional with simple ui.

Acknowledgments

Give credit to contributors or external resources that aided in the development of the system

This system reference site link:

- https://www.laravelia.com/post/laravel-10-roles-and-permissions-without-package
- https://www.laravelia.com/post/laravel-10-gate-and-policy-tutorial-with-example
- https://www.devopsschool.com/blog/how-to-create-multiple-role-based-authentication-and-access-control-in-laravel-application/
- https://dev.to/zaxwebs/implementing-simple-role-based-authorization-in-laravel-8-using-gates-1n44
- https://step2.dev/en/blog/7-laravel-gates-policies-guards-explained
- https://chat.openai.com/c/3d416b34-34cc-46ad-84f1-1dbf66ccec36