Final Project Exploration Approval

Acknowledgment Statement for Final Project Exploration Approval

I, William Maddock, acknowledge and accept the objectives and user stories outlined in this document for my final project exploration. I am committed to exploring the implementation of role-based access control and security clearance features, as well as addressing any challenges that arise in developing this solution to enhance the functionality and user experience for Shipping Agents and other stakeholders.

Description of What You Want to Explore and Learn

I want to explore and implement a security clearance feature that ensures users with specific roles, such as Shipping Agents, have the necessary permissions to access restricted areas like elevators and stairwells. My goal is to learn more about implementing role-based access control within a Rails application and to manage complex permissions dynamically based on user roles and current location requirements.

This feature should also include real-time feedback to inform users if they lack the required access permissions before they attempt entry, ensuring smooth and efficient workflows for all users.

User Story for Implementation

User Story: Prevent Security Clearance Issues for Access to Restricted Areas

As a Shipping Agent

I want to avoid being stuck in elevators or stairwells due to security clearance issues

So that I can efficiently complete my deliveries without delays.

Acceptance Criteria:

1. Shipping agents have the necessary clearance for elevator and stairwell access.
2. The system checks clearance for the building and delivery floors before access is granted.
3. Shipping agents are notified of any clearance issues before attempting access (Logistic Manager “approved or denied” the shipping agents elevated request).

Resources you found that will help you learn and implement

Here are some resources to help you implement dark mode in web applications, including approaches for Rails applications:

1. MDN Web Docs: Prefers Color Scheme Media Query

> **MDN on prefers-color-scheme** - Learn about the CSS media query prefers-color-scheme to automatically detect users' color scheme preferences (light or dark mode). This guide is a great introduction to how to set up dark mode purely with CSS.

2. CSS Tricks: Dark Mode in CSS

> **CSS Tricks Guide on Dark Mode** - This guide offers a comprehensive look at implementing dark mode in web applications, covering CSS, JavaScript toggles, and the prefers-color-scheme media query.

3. Rails Guides: JavaScript and CSS for Dynamic Mode Switching

> **Rails Asset Pipeline Guide** - For Rails applications, you can dynamically serve CSS for light and dark mode by managing CSS files with the Rails Asset Pipeline. Consider separate CSS files for each theme and use JavaScript to toggle them.
> **Stimulus Reflex Dark Mode Tutorial** - If you're using Stimulus JS with Rails, this tutorial demonstrates how to add a dark mode toggle to a Rails app using Stimulus.

4. Using Tailwind CSS for Dark Mode

> **Tailwind Dark Mode Guide** - Tailwind CSS offers a built-in dark mode feature that uses the prefers-color-scheme media query. This can be useful if you're using Tailwind for styling in Rails and want a straightforward way to implement dark mode.

5. JavaScript Toggle with Local Storage

> **Toggling Dark Mode with Local Storage (CSS-Tricks)** - This tutorial provides a method for creating a dark mode toggle using JavaScript and storing the user's preference with Local Storage so that it persists across sessions.

6. Dark Mode Icon Library for Visual Cues

> **Heroicons** - Heroicons offers a collection of icons, including light and dark mode icons, that you can use to indicate dark mode toggling. This is especially useful in Rails for adding visual cues for the toggle.

7. Frameworks and Libraries Supporting Dark Mode

Bootstrap Dark Mode - Bootstrap 5 includes dark mode utilities that make it easier to add dark mode to your Rails app if you are using Bootstrap as your CSS framework.
Material UI Dark Mode - Material UI provides dark mode support if you are using it with React. You can still adapt the Material Design color principles for Rails applications.

8. Dark Mode Design Principles

Dark Mode Design Best Practices by Apple - Apple's Human Interface Guidelines provide useful design principles to consider when implementing dark mode, such as maintaining contrast and adjusting colors for readability.

9. Articles and Best Practices

Building Dark Mode: Tips from Smashing Magazine - This article offers tips on building dark mode in web apps, including common pitfalls, color adjustments, and best practices.
Google Material Design Dark Mode - Google's Material Design guidelines for dark theme provide recommendations on colors, elevation, and contrast that ensure accessibility and readability.

Here are some helpful resources for working with databases in Rails 7:

1. Rails 7 Active Record Basics

Active Record Basics Guide - The official Rails guide provides an excellent introduction to Active Record, Rails' ORM layer, including how to interact with your database, model creation, and querying basics.

2. Migrations in Rails 7

Active Record Migrations Guide - This guide covers database migrations in detail, including creating and running migrations, modifying tables, and managing schema changes.

3. Active Record Query Interface

Active Record Querying Guide - Learn how to query the database using Rails' Active Record Query Interface. This guide covers common queries, complex conditions, joins, and scopes.

4. Rails 7 Active Record Associations

Active Record Associations Guide - This guide goes in-depth on defining associations in models, such as belongs_to, has_many, has_one, and has_many :through, helping you manage relationships between models in Rails.

5. Using Rails Console for Database Interactions

Rails Console Guide - The Rails console is an invaluable tool for testing and debugging queries directly within Rails. This guide covers how to start and use the Rails console effectively for testing database queries.

6. Active Record Callbacks

Active Record Callbacks Guide - Learn how to define hooks into the lifecycle of Active Record models, such as before_save, after_create, and after_update.

7. Schema Management

Schema and Structure Files - Learn about schema.rb and structure.sql for keeping track of the database schema and versioning. This is essential for maintaining database integrity across different environments.

8. Working with Encrypted and Sensitive Data in Rails 7

Rails 7 Encryption Guide - Rails 7 introduced built-in Active Record encryption. This guide covers how to securely store and manage sensitive data within your database.

9. Rails 7 and PostgreSQL

Rails and PostgreSQL - For those using PostgreSQL, this section of the configuration guide provides details on setting up and configuring PostgreSQL in Rails, including environment-specific configurations.

10. Database Maintenance and Optimization

Database Indexes - Learn how to use indexes in Rails to optimize database queries, improve performance, and ensure efficient data retrieval.
Rails Caching Guide - For large datasets, Rails offers a caching framework that integrates with Active Record to speed up database-heavy operations.

11. Testing with Active Record

Rails Testing Guide - This guide covers testing models, migrations, and interactions with the database in Rails 7, which is essential for ensuring the stability of your application as you make changes.