

Title: Container Management System

Description: The objective is to develop a Secure Container Management System that allows authenticated users to manage and track container shipments, and handle PDF file uploads related to container documentation. Ensure that the file upload functionality enforces a maximum file size of 5 MB and only accepts PDF files. Implement authentication and authorization to control access to the application.

Requirements:

1. Create a new ASP.NET Core MVC project with user authentication and authorization.
2. Implement login functionality using ASP.NET Core Identity or a similar authentication framework.
3. Set up a connection to an Oracle database to store container information, including details such as container numbers, shipment dates, origins, destinations, and status.
4. Design a model for container data and ensure that it includes the necessary fields:
 - Container ID (auto-generated)
 - Container Number
 - Shipment Date
 - Origin Port
 - Destination Port
 - Status (e.g., In Transit, At Dock, Delivered)
 - File Upload (for container documentation)
5. Create a page to add new container details to the system. This page should include a form with fields for the container's information and the ability to upload a PDF file as their container documentation.
6. Implement server-side validation to ensure that all required fields are filled out correctly when adding a new container.
7. Implement file upload functionality, allowing the user to select and upload a PDF file with a maximum size of 5 MB. Save the uploaded file to a folder on the server, and store the file path in the database.
8. Create a page to view a list of containers with their basic information and a link to download their associated container documentation.

9. Implement a dashboard that displays a list of containers, like a tracking system. The dashboard should allow authenticated users to:
 - View container details.
 - Filter containers by status.
10. Implement a jQuery-based filtering mechanism that allows users to filter containers by status on the dashboard without reloading the page.
11. Create a navigation menu for easy access to the dashboard, and container management features.
12. Implement error handling and validation for file uploads, ensuring that only PDF files are accepted and handling any potential errors gracefully.
13. Implement proper error handling and validation throughout the application.
14. Use proper coding practices and conventions, including separation of concerns and proper error handling.