Employee Management System (EMS)

Table of Contents

Project Overview

The **Employee Management System (EMS)** is a web application designed to manage employee data efficiently. It allows an admin user to perform Create, Read, Update, and Delete (CRUD) operations on user and employee entities. The application leverages modern technologies for a seamless user experience and robust backend functionality.

Technologies Used

Backend

- **Framework**: .NET 8.0
- **Packages**:
    - `AutoMapper.Extensions.Microsoft.DependencyInjection` (v12.0.1)
    - `BCrypt.Net-Next` (v4.0.3)
    - `Microsoft.AspNetCore.Authentication.JwtBearer` (v8.0.8)

- `Microsoft.EntityFrameworkCore` (v8.0.8)
- `Microsoft.EntityFrameworkCore.Design` (v8.0.8)
- `Microsoft.EntityFrameworkCore.SqlServer` (v8.0.8)
- `Swashbuckle.AspNetCore` (v6.4.0)

Frontend

- **Framework**: React (v18.3.1)
- **Packages**:
  - `@mui/material (v5.15.10)`
  - `@mui/x-data-grid (v6.19.5)`
  - `@reduxjs/toolkit (v2.2.1)`
  - `axios (v1.6.5)`
  - `jwt-decode (v4.0.0)`
  - `react-error-boundary (v4.0.12)`
  - `react-hook-form (v7.50.0)`
  - `react-icons (v5.0.1)`
  - `react-redux (v9.1.0)`
  - `react-router-dom (v6.21.2)`
  - `react-toastify (v10.0.4)`
  - `zod (v3.22.4)`

Database

- **Database Management**: Microsoft SQL Server
- **Version**: SQL Management Studio 18.12.1

Architecture

The SPA application follows a feature-based architecture, which organizes the codebase by feature rather than by type. This makes it easier to manage and scale as new features are added. The backend uses the Repository Pattern to separate data access logic, improving maintainability and testability.

Installation

Clone the Repository:

https://github.com/imrushikesh/EMS/

```
git clone <repository-url>

 EMS_API Branch - for .NET

EMS_SPA Branch - for React SPA

Please Clone each branch in a separate folder and then run.
```

1. **Backend Setup**:
    - If Needed Update the connection string in `appsettings.json` to point to your SQL Server database
    - Run the EMS_DB script from MAIN branch
2. **Frontend Setup**:
    - Install Node globally from the official Website.
    - Clone the branch EMS_SPA then Navigate to the frontend project directory
    - Install the required dependencies: command in terminal - `npm install`
    - To Run the project in terminal - `npm start`

Authentication

The EMS application employs JWT (JSON Web Token) for user authentication. The admin user can log in using the following credentials:

- **Username**: Admin
- **Password**: Admin@123

Upon successful login, the user will be redirected to the User management page.

Entities

1. User

- Represents the users of the system.
- Admin user cannot be edited or deleted.

2. Employee

- Represents employee data.
- Allows CRUD operations.

CRUD Operations

User Management

- **Create**: Add new users.
- **Read**: View existing users.
- **Update**: Edit user details
- **Delete**: Remove users

Employee Management

- Similar CRUD operations for employee entities.

Usage

After logging in with admin credentials, users will land on the User management page. Here, they can:

- Read The list of Users.
- Add new users.
- Update existing users (except the Admin).

- Delete users.

To access the Employee management page, simply click the navigation button labeled "Employee."

Conclusion

The Employee Management System (EMS) is a comprehensive solution for managing user and employee data efficiently. With a robust backend and a user-friendly frontend, it offers essential functionalities while following best practices in software architecture. Future enhancements can include additional features.