# Employee Management System

## Requirements

- **Functional:**
  - **Employee Information Management:**
    - **List Employee:** The system should allow users to view a list of all employees, including their basic details (name, position, department, etc.).
    - **Add Employee:** Users should be able to add new employees by entering relevant information.
    - **Update Employee:** The system must allow users to edit employee details (e.g., contact information, job roles) as needed.
    - **Delete Employee:** Users should be able to remove an employee from the system.
    - **Pagination:** If there are many employees, the system should support pagination to display a manageable number of records per page.
    - **Sorting:** Users should be able to sort employee records based on different criteria (e.g., name, hire date, department).
  - **User Authentication:**
    - **Login:** Employees and administrators should be able to log in securely using their credentials.
    - **Registration:** New users (employees or managers) should be able to register for an account.
  - **Security and Access Control:**
    - **Role-Based Access:** Different user roles (e.g., employee, manager, HR) should have varying levels of access to features and data.
    - **Data Privacy:** Ensure that sensitive employee information is securely stored and accessible only to authorized users.
  - **Logout:**
    - Users should be able to log out of the system securely.

- **Non-Functional:**
  - **Usability:**

    - **User-Friendly Interface:** The application should have an intuitive and easy-to-navigate interface.
    - **Responsive Design:** Ensure that the system works well on different devices (desktops, tablets, mobile phones).
  - **Performance:**
    - **Response Time:** The system should respond quickly to user requests (e.g., loading employee lists, updating records).
    - **Scalability:** The application should handle a growing number of employees without performance degradation.
  - **Reliability:**
    - **Availability:** The system should be available for use during business hours.
  - **Security:**
    - **Authentication and Authorization:** Ensure secure login and role-based access control.
    - **Data Encryption:** Sensitive data (such as passwords) should be encrypted.
    - **Protection Against Attacks:** Implement security measures to prevent unauthorized access or attacks (e.g., SQL injection).
  - **Maintainability:**
    - **Code Quality:** Develop clean, well-documented code to facilitate future maintenance.
    - **Modularity:** Design the system in a modular way to allow for easy updates and enhancements.