**Employee Management System - Documentation**

**1. Introduction**

The **Employee Management System** is a web-based application that allows users to manage employee records, including creating, retrieving, updating, and deleting employee details. The backend is built using **Spring Boot, MySQL, Hibernate, and JPA**, while the frontend is developed using **Ext JS**.

**2. Technology Stack**

**Backend (Spring Boot):**

* Java 17
* Spring Boot 3.x
* Spring Data JPA (Hibernate)
* MySQL Database
* Maven
* CORS Configuration

**Frontend (Ext JS):**

* Ext JS 7.8
* Sencha Cmd 7.8
* JavaScript (ES6)
* AJAX Requests

**3. Project Structure**

**Backend - Spring Boot**

* Controller Layer: Handles HTTP requests.
* Service Layer: Contains business logic.
* Repository Layer: Manages database interactions.
* Model Layer: Defines entity structures.
* Configuration Layer: Configures application settings.

**Frontend - Ext JS**

* Model: Defines the employee data structure.
* Store: Manages data retrieval from the backend.
* View: Displays the employee list in a grid.
* Main Application: Initializes the Ext JS framework.

**4. Backend Implementation**

* Defines entity classes for employees.
* Uses JPA repositories for database operations.
* Exposes RESTful APIs for CRUD operations.
* Implements CORS configuration to allow frontend access.
* Runs on a local server with MySQL as the database.

**5. Frontend Implementation**

* Fetches employee data from the backend API.
* Displays employee details in a grid format.
* Uses AJAX to communicate with the backend.
* Auto-loads employee records on page load.
* Provides user interface components for easy navigation.

**6. Summary**

* The backend exposes REST APIs for employee management.
* The frontend retrieves and displays employee data in a structured format.
* CORS is enabled to allow cross-origin requests.
* The system allows CRUD operations for employee records.