Employee Management System – Project Report

## Title

Employee Management System Using Spring Boot and MySQL

## Overview

This backend system manages employee data using Spring Boot and MySQL. It offers REST APIs for CRUD operations, filtering, pagination, sorting, and bulk data operations. The project is built with clean architecture, DTOs, input validation, and exception handling — reflecting real-world development practices.

## Objectives

* Perform full CRUD on employee records.
* Support filter by role and salary range.
* Enable pagination and sorting of employee lists.
* Provide bulk insert and status toggle.
* Use validation, DTOs, and global error handling.
* Document APIs using Swagger UI.

## Tech Stack

* Java 17
* Spring Boot 3.x
* Spring Data JPA
* MySQL
* Lombok
* Swagger UI

## Key Features

|  |  |
| --- | --- |
| **Feature** | **Description** |
| CRUD APIs | Add, read, update, delete employees |
| Role Filter | Get employees by job role |
| Salary Filter | Filter by min and max salary |
| Pagination + Sort | /page?page=0&size=5&sort=name |
| Toggle Status | Activate or deactivate employees |
| Bulk Insert | Add multiple employees at once |
| Average Salary | Get avg. salary by role |
| Validation + DTO | Clean data flow with bean validation |
| Exception Handling | Centralized error response |
| Swagger UI | API testing & documentation |

## API Endpoints Summary

* POST /api/employees
* GET /api/employees
* GET /api/employees/{id}
* PUT /api/employees/{id}
* DELETE /api/employees/{id}
* GET /api/employees/role/{role}
* GET /api/employees/salary?min=40000&max=100000
* GET /api/employees/page?page=0&size=5&sort=name
* POST /api/employees/bulk
* PUT /api/employees/{id}/status?active=true
* GET /api/employees/active
* GET /api/employees/average-salary/{role}

**Swagger URL:** <http://localhost:8080/swagger-ui.html>

## Conclusion

This project demonstrates backend engineering using modern Spring Boot features. It meets internship standards and is excellent for showcasing skills in REST APIs, MySQL, JPA, and real-world backend development.