**Exercise 6: Employee Management System - Implementing Pagination and Sorting**

**Business Scenario:**

**Add pagination and sorting capabilities to your employee search functionality.**

**Instructions:**

1. **Pagination:**
   * **Implement pagination for the employee list using Page and Pageable.**
2. **Sorting:**
   * **Add sorting functionality to your queries.**
   * **Combine pagination and sorting in your search endpoint.**

**1. Pagination**

Pagination is a crucial feature in applications that deal with large datasets, allowing you to retrieve and display data in chunks instead of loading everything at once. Spring Data JPA provides built-in support for pagination using the Page and Pageable interfaces.

**Implementing Pagination**

You can add pagination to your repository methods by including Pageable as a parameter. The Pageable interface allows you to specify the page number, page size, and sort order.

**Pagination in the Controller**

To implement pagination in your controller, you can use Spring’s Pageable interface, which allows you to pass the page number and size as request parameters.

**2. Sorting**

Sorting enables the organization of data in ascending or descending order based on one or more fields. Spring Data JPA provides a Sort class that can be used in conjunction with Pageable to achieve this.

**Implementing Sorting**

Sorting can be combined with pagination by adding a Sort object to the Pageable parameter.