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

**EmployeeController**

package com.employee.management.system.controllers;

import com.employee.management.system.entities.Employee;

import com.employee.management.system.repositories.EmployeeRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.data.domain.Sort;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/employees")

public class EmployeeController {

@Autowired

private EmployeeRepository employeeRepository;

@GetMapping

public Page<Employee> getAllEmployees(Pageable pageable) {

return employeeRepository.findAll(pageable);

}

@GetMapping("/search")

public Page<Employee> searchEmployees(@RequestParam String name, Pageable pageable) {

return employeeRepository.findEmployeesByNameContaining(name, pageable);

}

}

**EmployeeRepository**

package com.employee.management.system.repositories;

import com.employee.management.system.entities.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

@Query("SELECT e FROM Employee e WHERE e.name LIKE %:name%")

Page<Employee> findEmployeesByNameContaining(@Param("name") String name, Pageable pageable);

}

In the above code, I have implemented pagination and sorting capabilities in the EmployeeController and EmployeeRepository.

I have used Spring Data JPA's Page and Pageable classes to implement pagination, and Sort class to implement sorting.

The getAllEmployees method in EmployeeController now returns a Page of employees, and the searchEmployees method takes a Pageable parameter to allow pagination and sorting.

The findEmployeesByNameContaining method in EmployeeRepository now returns a Page of employees and takes a Pageable parameter to allow pagination and sorting.

Note that I have used @Query annotation to define a custom query method in EmployeeRepository, and Pageable parameter to enable pagination and sorting.