**EmployeeRepository.java**

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.model.Employee;

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

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Inherits built-in paging and sorting support from JpaRepository

}

**EmployeeService.java**

package com.example.employeemanagementsystem.service;

import com.example.employeemanagementsystem.model.Employee;

import com.example.employeemanagementsystem.repository.EmployeeRepository;

import lombok.RequiredArgsConstructor;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.stereotype.Service;

@Service

@RequiredArgsConstructor

public class EmployeeService {

private final EmployeeRepository employeeRepository;

public Page<Employee> getAllEmployees(Pageable pageable) {

return employeeRepository.findAll(pageable);

}

// Other CRUD methods from earlier exercise...

}

**EmployeeController.java:**

package com.example.employeemanagementsystem.controller;

import com.example.employeemanagementsystem.model.Employee;

import com.example.employeemanagementsystem.service.EmployeeService;

import lombok.RequiredArgsConstructor;

import org.springframework.data.domain.\*;

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

@RestController

@RequestMapping("/api/employees")

@RequiredArgsConstructor

public class EmployeeController {

private final EmployeeService employeeService;

// Endpoint with pagination and sorting

@GetMapping("/paginated")

public Page<Employee> getPaginatedEmployees(

@RequestParam(defaultValue = "0") int page,

@RequestParam(defaultValue = "5") int size,

@RequestParam(defaultValue = "id") String sortBy,

@RequestParam(defaultValue = "asc") String order

) {

Sort sort = order.equalsIgnoreCase("desc")

? Sort.by(sortBy).descending()

: Sort.by(sortBy).ascending();

Pageable pageable = PageRequest.of(page, size, sort);

return employeeService.getAllEmployees(pageable);

}

// Other CRUD endpoints from earlier exercise...

}