**WEEK 3**

**EXERCISE 3**

**Employee Management System - Creating Repositories**

**1.Overview of Spring Data Repositories:**

**Benefits of Using Spring Data Repositories:**

* Simplified Data Access: Spring Data repositories provide a simplified way to perform CRUD operations and complex queries without the need to write boilerplate code.
* Automatic Implementation: Interfaces extending JpaRepository are automatically implemented by Spring Data, reducing the need for manual coding.
* Derived Queries: You can define query methods by simply naming them according to the properties of the entity, and Spring Data will automatically implement them.
* Pagination and Sorting: Repositories support pagination and sorting out-of-the-box.

**2. Creating Repositories:**

**a. EmployeeRepository Interface**

* **Purpose:** Provides CRUD operations for Employee entities.

**EmployeeRepository Code:**

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.entity.Employee;

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

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

List<Employee> findByName(String name);

Employee findByEmail(String email);

}

**b. DepartmentRepository Interface**

* **Purpose:** Provides CRUD operations for Department entities.

**DepartmentRepository Code:**

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.entity.Department;

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

import org.springframework.stereotype.Repository;

@Repository

public interface DepartmentRepository extends JpaRepository<Department, Long> {

// Derived query method to find departments by name

Department findByName(String name);

}