**Spring Data JPA with Spring Boot, Hibernate**

**EXERCISE 4: Difference between JPA, Hibernate and Spring Data JPA**

**Source Code**

**Using Hibernate (JPA implementation):**

public Integer addEmployee(Employee employee) {

Session session = factory.openSession(); // factory is Hibernate's SessionFactory

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee); // Manual save

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

EmployeeRepository.java:

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

import org.springframework.stereotype.Repository;

import com.example.model.Employee;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java:

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee); // Just one line

}

public List<Employee> getAllEmployees() {

return employeeRepository.findAll(); // Again, one line!

}

}

Application.java:

@SpringBootApplication

public class Application {

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(Application.class, args);

EmployeeService service = context.getBean(EmployeeService.class);

Employee emp = new Employee(1, "Charitha", "Developer");

service.addEmployee(emp);

System.out.println("All Employees: " + service.getAllEmployees());

}

}

**Output**

