**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**

* **Java Persistence API (JPA):**

JSR 338 Specification for persisting, reading and managing data from Java objects. Does not contain concrete implementation of the specification. Hibernate is one of the implementation of JPA.

* **Hibernate:**

ORM Tool that implements JPA.

* **Spring Data JPA:**

Does not have JPA implementation, but reduces boiler plate code. This is another level of abstraction over JPA implementation provider like Hibernate. Manages transactions.

* **Hibernate Example:**

// Method to CREATE an employee in the database

public Integer addEmployee(Employee employee) {

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

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

tx.commit();

} catch (HibernateException e) {

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

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

* **Spring Data JPA Example:**
  + **EmployeeRespository.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;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}