**Exercise 7: Test Custom Repository Query**

**Task:** Add and test a custom query method

user-repository-demo/

├── pom.xml

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── example/

│ │ │ └── demo/

│ │ │ ├── DemoApplication.java

│ │ │ ├── model/

│ │ │ │ └── User.java

│ │ │ └── repository/

│ │ │ └── UserRepository.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └── static/

│ │ └── templates/

│

│ └── test/

│ └── java/

│ └── com/

│ └── example/

│ └── demo/

│ └── UserRepositoryTest.java

**UserRepository.java :-**

package com.example.demo.repository;

import com.example.demo.model.User;

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

import java.util.List;

public interface UserRepository extends JpaRepository<User, Long> {

List<User> findByName(String name); // Custom method

}

**User.java :-**

package com.example.demo.model;

import jakarta.persistence.\*;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

// Getters and setters

}

**USerRepositoryTest.java :-**

package com.example.demo;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTest;

import java.util.List;

import static org.assertj.core.api.Assertions.assertThat;

@DataJpaTest

public class UserRepositoryTest {

@Autowired

private UserRepository userRepository;

@BeforeEach

void setUp() {

userRepository.deleteAll();

User u1 = new User();

u1.setName("Naresh");

u1.setEmail("naresh1@example.com");

User u2 = new User();

u2.setName("Naresh");

u2.setEmail("naresh2@example.com");

User u3 = new User();

u3.setName("Ramesh");

u3.setEmail("ramesh@example.com");

userRepository.saveAll(List.of(u1, u2, u3));

}

@Test

public void testFindByName() {

List<User> nareshUsers = userRepository.findByName("Naresh");

assertThat(nareshUsers).hasSize(2);

assertThat(nareshUsers.get(0).getEmail()).contains("naresh");

assertThat(nareshUsers.get(1).getEmail()).contains("naresh");

}

}

**pom.xml :-**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>user-repository-demo</artifactId>

<version>1.0.0</version>

<properties>

<java.version>17</java.version>

<spring.boot.version>3.2.0</spring.boot.version>

</properties>

<dependencies>

<!-- Spring Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 In-memory DB -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Test -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- Jackson (optional) -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Plugin -->

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>${spring.boot.version}</version>

</plugin>

<!-- Compiler Plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

<compilerArgs>

<arg>-parameters</arg>

</compilerArgs>

</configuration>

</plugin>

</plugins>

</build>

</project>

Output :-

