**Exercise 5.2: Mocking a Repository in a Service Test**

**CODE:**

**User Class (Entity): -**

package org.example;  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
  
@Entity  
public class User {  
 @Id  
 private int id;  
 private String name;  
  
 public User() {  
 }  
  
 public User(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**UserRepository Interface: -**

package org.example;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface UserRepository extends JpaRepository<User,Integer> {  
}

**Service Class: -**

package org.example;  
public class Service {  
 private final UserRepository userRepository;  
  
 public Service(UserRepository userRepository) {  
 this.userRepository = userRepository;  
 }  
  
 public User getUserById(int id) {  
 return userRepository.findById(id).orElse(null);  
 }  
  
}

**ServiceTest Class: -**

package org.example;  
import org.junit.jupiter.api.Test;  
import org.junit.jupiter.api.extension.ExtendWith;  
import org.mockito.InjectMocks;  
import org.mockito.Mock;  
import org.mockito.junit.jupiter.MockitoExtension;  
import java.util.Optional;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
@ExtendWith(MockitoExtension.class)  
public class ServiceTest {  
  
 @Mock  
 private UserRepository userRepository;  
  
 @InjectMocks  
 private Service service;  
  
 @Test  
 public void mockingRepoServiceTest() {  
 User mockUser = new User(1, "Alice Reddy");  
 *when*(userRepository.findById(1)).thenReturn(Optional.*of*((mockUser)));  
  
 User result = service.getUserById(1);  
 *assertEquals*("Alice Reddy", result.getName());  
 System.*out*.println("User name is: "+result.getName());  
 }  
}

**OUTPUT:**

