**CODE:**

package com.example;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class User {

@Id

private Long id;

private String name;

public User() {}

public User(Long id, String name) {

this.id = id;

this.name = name;

}

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

package com.example;

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

public interface UserRepository extends JpaRepository<User, Long> {

}

package com.example;

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

import org.springframework.stereotype.Service;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public User getUserById(Long id) {

return userRepository.findById(id).orElse(null);

}

}

package com.example;

import org.junit.jupiter.api.Test;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.MockitoAnnotations;

import java.util.Optional;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.when;

class UserServiceTest {

@Mock

private UserRepository userRepository;

@InjectMocks

private UserService userService;

public UserServiceTest() {

MockitoAnnotations.openMocks(this); // initialize mocks

}

@Test

void testGetUserById\_ReturnsUser() {

Long userId = 1L;

User mockUser = new User(userId, "Sabarna");

when(userRepository.findById(userId)).thenReturn(Optional.of(mockUser));

User result = userService.getUserById(userId);

assertEquals("Sabarna", result.getName());

assertEquals(userId, result.getId());

}

@Test

void testGetUserById\_UserNotFound() {

Long userId = 2L;

when(userRepository.findById(userId)).thenReturn(Optional.empty());

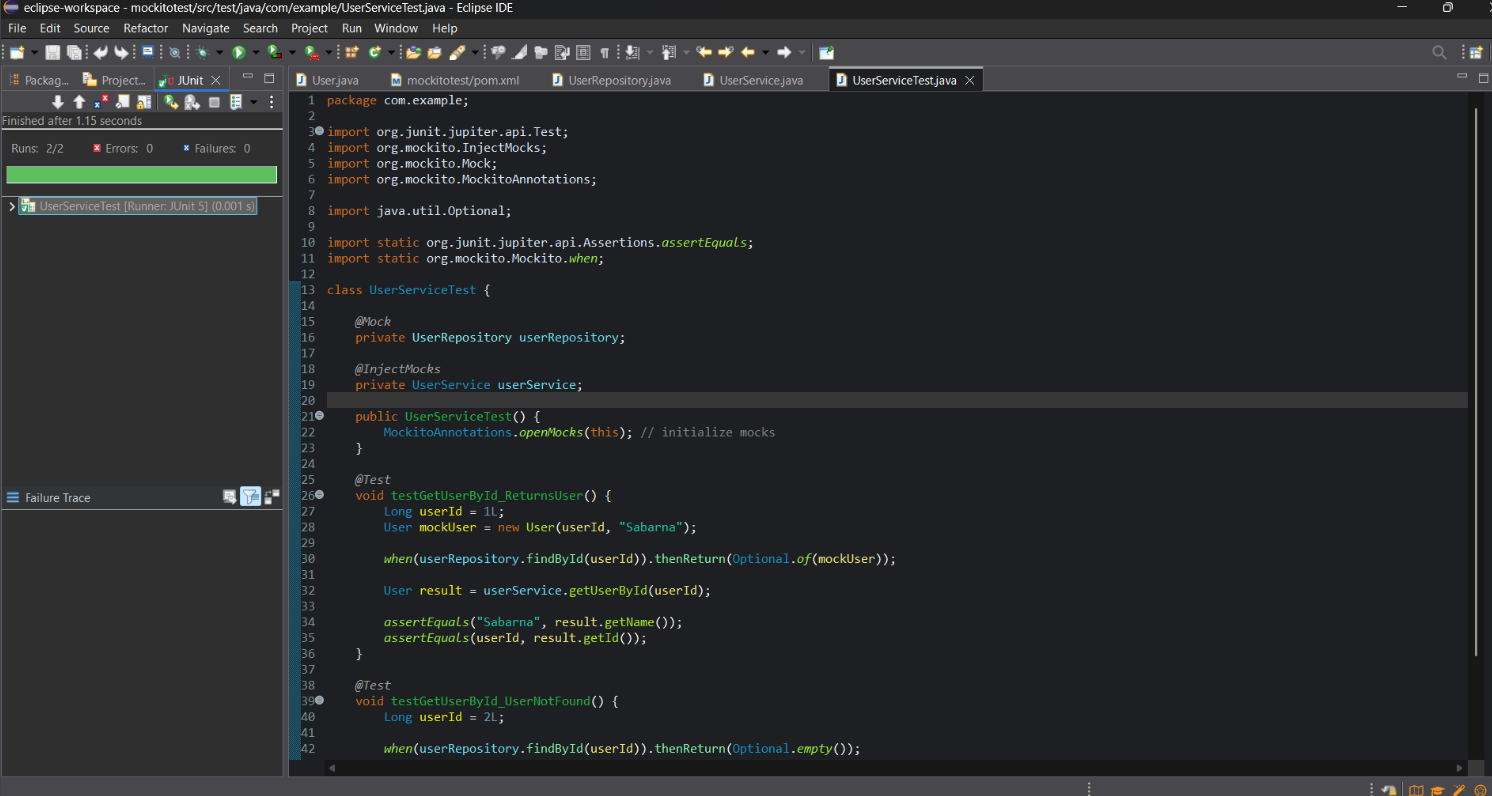
User result = userService.getUserById(userId);

assertEquals(null, result);

}

}

**OUTPUT:**

****