**CODE:**

package com.example.demo.controller;

import com.example.demo.entity.User;

import com.example.demo.service.UserService;

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

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/users")

public class UserController {

@Autowired

private UserService userService;

@GetMapping("/{id}")

public ResponseEntity<User> getUser(@PathVariable Long id) {

return ResponseEntity.ok(userService.getUserById(id));

}

}

package com.example.demo.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class User {

@Id

private Long id;

private String name;

// Getters and Setters

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.demo.repository;

import com.example.demo.entity.User;

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

public interface UserRepository extends JpaRepository<User, Long> {

}

package com.example.demo.service;

import com.example.demo.entity.User;

import com.example.demo.repository.UserRepository;

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.demo.controller;

import com.example.demo.entity.User;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.MockitoAnnotations;

import org.springframework.http.ResponseEntity;

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

import static org.mockito.Mockito.\*;

public class UserControllerTest {

@Mock

private UserService userService;

@InjectMocks

private UserController userController;

public UserControllerTest() {

MockitoAnnotations.openMocks(this); // Initialize mocks

}

@Test

public void testGetUserById() {

System.out.println("✅ testGetUserById started...");

// Arrange

User mockUser = new User();

mockUser.setId(1L);

mockUser.setName("Sabarna");

when(userService.getUserById(1L)).thenReturn(mockUser);

// Act

ResponseEntity<User> response = userController.getUser(1L);

// Assert

System.out.println("Returned Name: " + response.getBody().getName());

assertEquals("Sabarna", response.getBody().getName());

assertEquals(1L, response.getBody().getId());

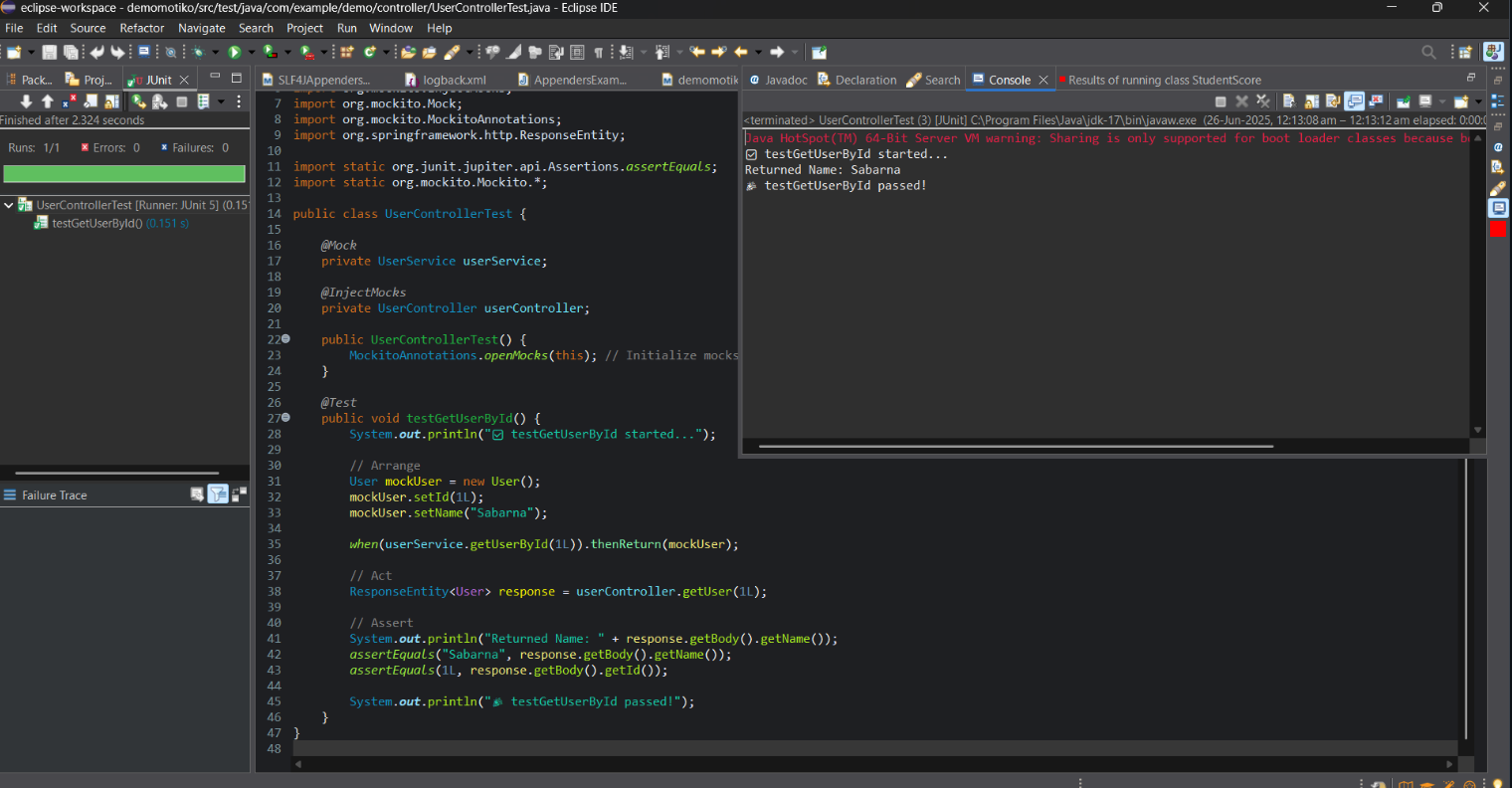
System.out.println("🎉 testGetUserById passed!");

}

}

}

**OUTPUT:**

****