**CODE:**

package com.example.demo.exception;

@SuppressWarnings("serial")

public class UserNotFoundException extends RuntimeException {

public UserNotFoundException(String message) {

super(message);

}

}

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "users")

public class User {

@Id

private Long id;

private String name;

public User() {}

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.model.User;

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

public interface UserRepository extends JpaRepository<User, Long> {

}

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

package com.example.demo;

import com.example.demo.exception.UserNotFoundException;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.context.SpringBootTest;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

public class UserServiceTest {

@Autowired

private UserService userService;

@Test

public void testUserNotFoundThrowsException() {

Long fakeId = 100L;

Exception exception = assertThrows(UserNotFoundException.class, () -> {

userService.getUserById(fakeId);

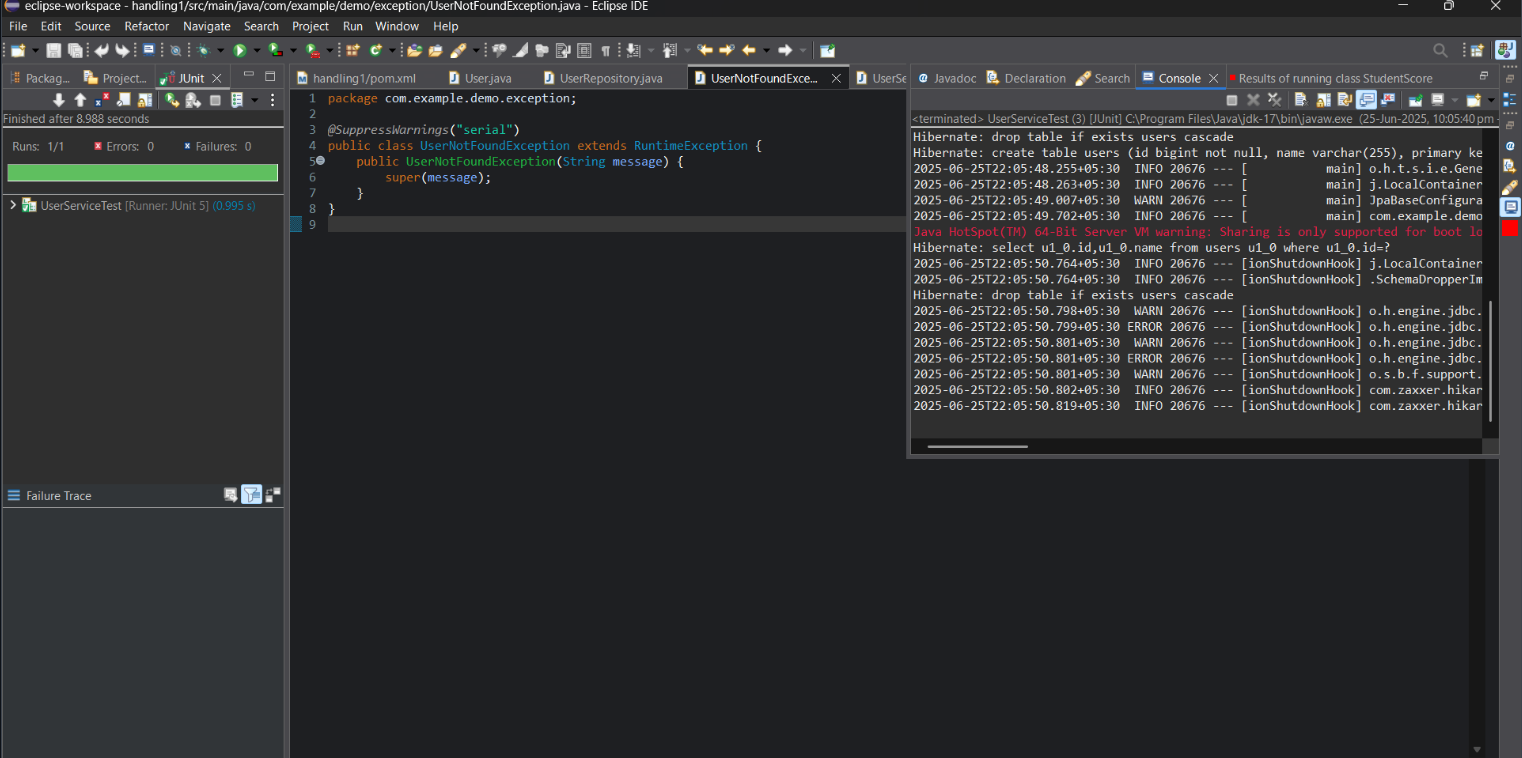
});

assertEquals("User not found with ID: 100", exception.getMessage());

}

}

**OUTPUT:**

****