**Exercise 5.8: Test Controller Exception Handling**

**CODE:**

**User Class(Entity): -**

package org.example;  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "users")  
public class User {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String name;  
 private String email;  
  
 public User() {}  
 public User(String name, String email) {  
 this.name = name;  
 this.email = email;  
 }  
  
 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; }  
  
 public String getEmail() { return email; }  
 public void setEmail(String email) { this.email = email; }  
}

**UserService Interface: -**

package org.example;  
  
public interface UserService {  
 User getUserById(Long id);  
}

**UserServiceImpl Class: -**

package org.example;  
import java.util.NoSuchElementException;  
  
public class UserServiceImpl implements UserService {  
  
 @Override  
 public User getUserById(Long id) {  
 throw new NoSuchElementException("No user with id " + id);  
 }  
}

**UserController Class: -**

package org.example;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping("/users")  
public class UserController {  
  
 private final UserService userService;  
  
 public UserController(UserService userService) {  
 this.userService = userService;  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<User> getUser(@PathVariable Long id) {  
 return ResponseEntity.*ok*(userService.getUserById(id));  
 }  
}

**UserControllerException Class: -**

package org.example;  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;  
import org.springframework.boot.test.mock.mockito.MockBean;  
import org.springframework.test.web.servlet.MockMvc;  
import java.util.NoSuchElementException;  
import static org.mockito.Mockito.*when*;  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*content*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
@WebMvcTest(controllers = UserController.class)  
public class UserControllerExceptionTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @MockBean  
 private UserService userService;  
  
 @Test  
 public void testUserNotFoundException() throws Exception {  
 Long userId = 1L;  
  
 *when*(userService.getUserById(userId)).thenThrow(new NoSuchElementException("User not found"));  
  
 mockMvc.perform(*get*("/users/{id}", userId))  
 .andExpect(*status*().isNotFound())  
 .andExpect(*content*().string("User not found"));  
  
 System.*out*.println("Exception handled correctly with 404 and message.");  
 }  
}

**GlobalExceptionHandler Class: -**

package org.example;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.ControllerAdvice;  
import org.springframework.web.bind.annotation.ExceptionHandler;  
import java.util.NoSuchElementException;  
  
@ControllerAdvice  
public class GlobalExceptionHandler {  
 @ExceptionHandler(NoSuchElementException.class)  
 public ResponseEntity<String> handleNotFound(NoSuchElementException ex) {  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).body("User not found");  
 }  
}

**OUTPUT:**

