**Exercise 5.3: Testing a REST Controller with MockMvc**

**CODE:**

**UserControllerTest 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.http.MediaType;  
import org.springframework.test.web.servlet.MockMvc;  
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.*jsonPath*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
@WebMvcTest(UserController.class)  
public class UserControllerTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @MockBean  
 private UserService userService;  
  
 @Test  
 public void testGetUserById() throws Exception {  
 // Arrange  
 User mockUser = new User(1L, "Alice Reddy", "alice@example.com");  
 *when*(userService.getUserById(1L)).thenReturn(mockUser);  
  
 // Act & Assert  
 mockMvc.perform(*get*("/users/1")  
 .accept(MediaType.*APPLICATION\_JSON*))  
 .andExpect(*status*().isOk())  
 .andExpect(*jsonPath*("$.id").value(1))  
 .andExpect(*jsonPath*("$.name").value("Alice Reddy"))  
 .andExpect(*jsonPath*("$.email").value("alice@example.com"));  
 System.*out*.println("User Details: \nName: "+mockUser.getName()+"\nEmail ID: "+mockUser.getEmail());  
 }  
}

**OUTPUT:**

