***Dương Ngọc Thảo – 1911066368***

***UNIT TEST CHỨC NĂNG ĐĂNG NHẬP WEBSITE***

* Unit test cho chức năng đăng nhâp ta cần xác định các bước cần thực hiện khi đăng nhập, bao gồm:

1. Nhập thông tin tài khoản (username và password).
2. Gửi thông tin tài khoản đến server.
3. Server xác thực thông tin tài khoản.
4. Nếu thông tin đúng, server cho phép đăng nhập và trả về kết quả thành công.
5. Nếu thông tin sai, server từ chối đăng nhập và trả về kết quả thất bại.

* Các test case để kiểm tra tính đúng đắn của chức năng đăng nhập.

1. Test case nhập đúng tài khoản và mật khẩu: kết quả trả về là true, người dùng được đăng nhập thành công.
2. Test case nhập sai tài khoản hoặc mật khẩu: kết quả trả về là false, người dùng không đăng nhập được.
3. Test case tài khoản hoặc mật khẩu để trống: kết quả trả về là false, người dùng không đăng nhập được.
4. Test case nhập thiếu thông tin tài khoản (chỉ nhập username hoặc password), kiểm tra kết quả trả về từ server.
5. Test case tài khoản bị khóa: kết quả trả về là false, người dùng không đăng nhập được.

* Viết unit test cho hàm đăng nhập với các test case đã tạo. Trong quá trình viết test case, ta cần sử dụng các phương thức của framework JUnit để kiểm tra kết quả trả về của hàm và so sánh với kết quả mong đợi.

Code:

import static org.junit.Assert.\*;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.Mockito;

import org.mockito.junit.MockitoJUnitRunner;

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.context.SecurityContextHolder;

@RunWith(MockitoJUnitRunner.class)

public class LoginControllerTest {

@InjectMocks

private LoginController loginController;

@Mock

private UserService userService;

@Test

public void testSuccessfulLogin() {

String username = "john\_doe";

String password = "password";

User user = new User(username, password);

Mockito.when(userService.authenticate(username, password)).thenReturn(user);

Authentication auth = new UsernamePasswordAuthenticationToken(username, password);

SecurityContextHolder.getContext().setAuthentication(auth);

String result = loginController.login(username, password);

assertEquals("redirect:/home", result);

}

@Test

public void testFailedLogin() {

String username = "jane\_doe";

String password = "password";

Mockito.when(userService.authenticate(username, password)).thenReturn(null);

String result = loginController.login(username, password);

assertEquals("login", result);

}

@Test

public void testMissingUsername() {

String username = "";

String password = "password";

String result = loginController.login(username, password);

assertEquals("login", result);

}

@Test

public void testMissingPassword() {

String username = "john\_doe";

String password = "";

String result = loginController.login(username, password);

assertEquals("login", result);

}

}