***Nguyễn Tứ: 1911066465***

**UNIT TEST: ĐĂNG KÝ TÀI KHOẢN**

Test case:

* Test case 1: Kiểm tra email không hợp lệ.
* Test case 2: Kiểm tra email đã được sử dụng trước đó.
* Test case 3: Kiểm tra mật khẩu ngắn hơn 8 ký tự.
* Test case 4: Kiểm tra mật khẩu không chứa chữ cái.
* Test case 5: Kiểm tra mật khẩu không chứa số.
* Test case 6: Kiểm tra họ tên để trống.
* Test case 7: Kiểm tra họ tên chứa ký tự đặc biệt.
* Test case 8: Kiểm tra địa chỉ để trống.
* Test case 9: Kiểm tra địa chỉ không chứa số nhà.
* Test case 10: Kiểm tra địa chỉ không chứa tên đường.

Code:

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class RegisterTest {

private Register register;

@Before

public void setUp() {

register = new Register();

}

@Test

public void testInvalidEmail() {

assertEquals("Invalid email format", register.register("invalidemail", "password123", "John Smith", "123 Main St"));

}

@Test

public void testExistingEmail() {

register.register("john@example.com", "password123", "John Smith", "123 Main St");

assertEquals("Email already exists", register.register("john@example.com", "password123", "John Smith", "123 Main St"));

}

@Test

public void testShortPassword() {

assertEquals("Password must be at least 8 characters long", register.register("john@example.com", "1234", "John Smith", "123 Main St"));

}

@Test

public void testPasswordWithoutLetter() {

assertEquals("Password must contain at least one letter", register.register("john@example.com", "12345678", "John Smith", "123 Main St"));

}

@Test

public void testPasswordWithoutNumber() {

assertEquals("Password must contain at least one number", register.register("john@example.com", "password", "John Smith", "123 Main St"));

}

@Test

public void testBlankName() {

assertEquals("Name cannot be blank", register.register("john@example.com", "password123", "", "123 Main St"));

}

@Test

public void testInvalidName() {

assertEquals("Name can only contain letters", register.register("john@example.com", "password123", "John Smith!", "123 Main St"));

}

@Test

public void testBlankAddress() {

assertEquals("Address cannot be blank", register.register("john@example.com", "password123", "John Smith", ""));

}

@Test

public void testAddressWithoutHouseNumber() {

assertEquals("Address must contain at least one house number", register.register("john@example.com", "password123", "John Smith", "Main St"));

}

@Test

public void testAddressWithoutStreetName() {

assertEquals("Address must contain at least one street name", register.register("john@example.com", "password123", "John Smith", "123"));

}

}