**UNIT TEST: QUÊN MẬT KHẨU**

**Test Case**

1. Kiểm tra người dùng có nhập email và email có đúng định dạng không.
2. Kiểm tra email có tồn tại trong hệ thống không.
3. Kiểm tra gửi email reset mật khẩu thành công hay không.
4. Kiểm tra tính mạnh của mật khẩu mới được nhập vào.
5. Kiểm tra lưu mật khẩu mới vào hệ thống thành công hay không.

**CODE:**

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

public class ForgotPasswordTest {

@Test

public void testResetPasswordEmptyEmail() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.resetPassword("");

Assertions.assertFalse(result);

Assertions.assertEquals("Email is required.", forgotPassword.getErrorMessage());

}

@Test

public void testResetPasswordInvalidEmail() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.resetPassword("invalidemail");

Assertions.assertFalse(result);

Assertions.assertEquals("Invalid email format.", forgotPassword.getErrorMessage());

}

@Test

public void testResetPasswordNonexistentEmail() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.resetPassword("nonexistentemail@example.com");

Assertions.assertFalse(result);

Assertions.assertEquals("Email does not exist in the system.", forgotPassword.getErrorMessage());

}

@Test

public void testResetPasswordValidEmail() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.resetPassword("validemail@example.com");

Assertions.assertTrue(result);

Assertions.assertNull(forgotPassword.getErrorMessage());

}

@Test

public void testCreateNewPasswordWeakPassword() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.createNewPassword("weak");

Assertions.assertFalse(result);

Assertions.assertEquals("Password must be at least 8 characters long.", forgotPassword.getErrorMessage());

}

@Test

public void testCreateNewPasswordStrongPassword() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.createNewPassword("strongpassword123");

Assertions.assertTrue(result);

Assertions.assertNull(forgotPassword.getErrorMessage());

}

@Test

public void testSaveNewPassword() {

ForgotPassword forgotPassword = new ForgotPassword();

boolean result = forgotPassword.saveNewPassword("newpassword");

Assertions.assertTrue(result);

Assertions.assertNull(forgotPassword.getErrorMessage());

}

}