

Trường: <b>ĐH CNTP TP.HCM</b> Khoa: <b>Công nghệ thông tin</b> Bộ môn: <b>Công nghệ phần mềm</b> MH: <b>TH Kiểm định phần mềm</b> MSMH:	<b>BÀI 4: Test UI</b>	
---	-----------------------	--

## A. MỤC TIÊU:

- Phân tích và thiết kế testcase cho kiểm tra chức năng.
- Xây dựng kịch bản kiểm chứng tự động black box.
- Biết các sử dụng Selenium IDE để thực hiện kiểm chứng tự động chức năng trên Website.

## B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

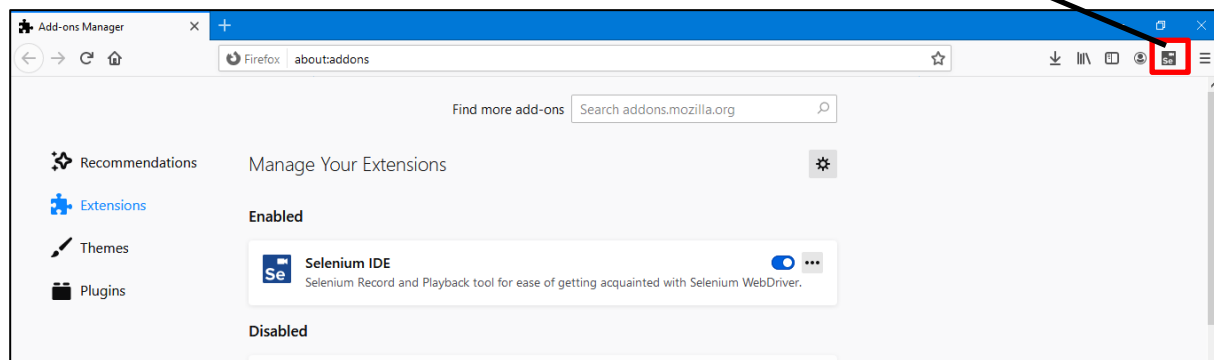
STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

## C. NỘI DUNG THỰC HÀNH

### I. Tóm tắt lý thuyết

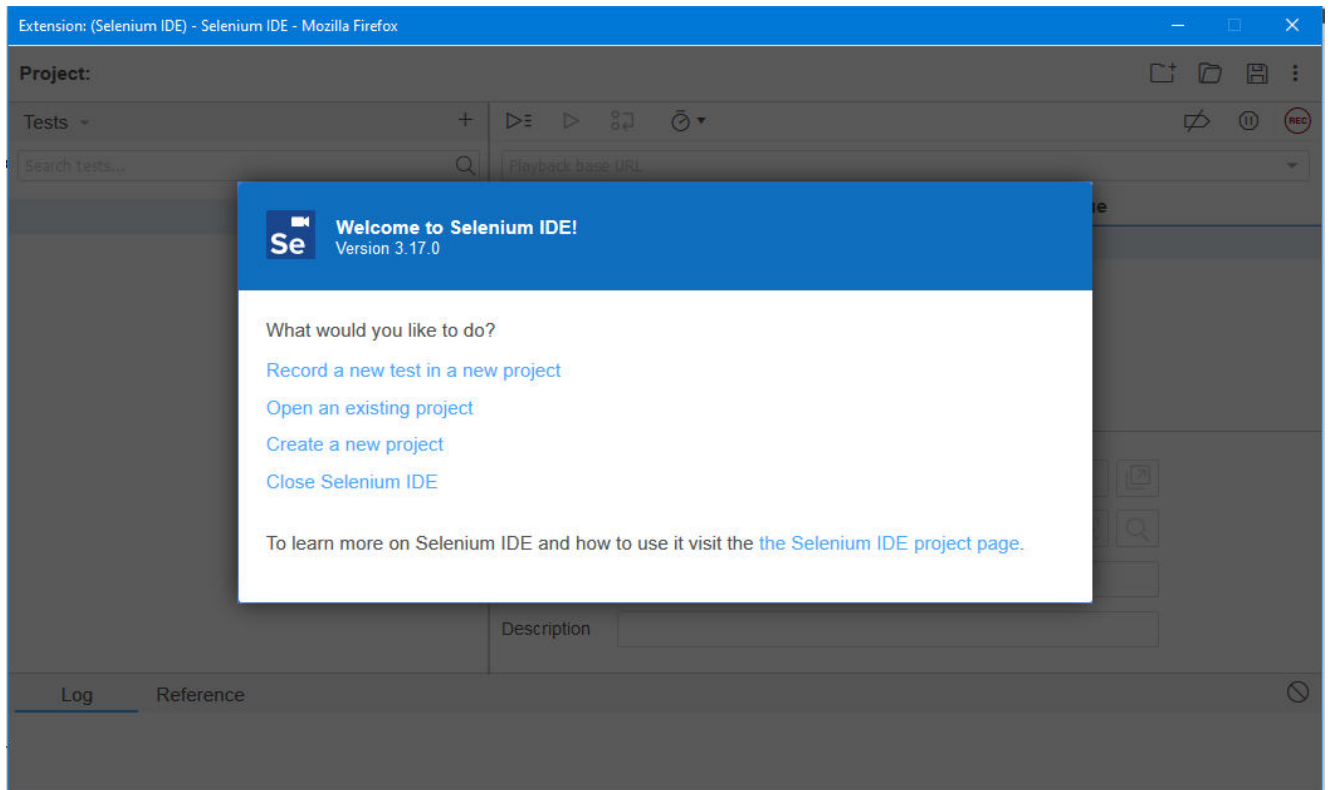
#### 1. Cài đặt Selenium vào Browser

##### Bước 1: Vào Add-on của Browser

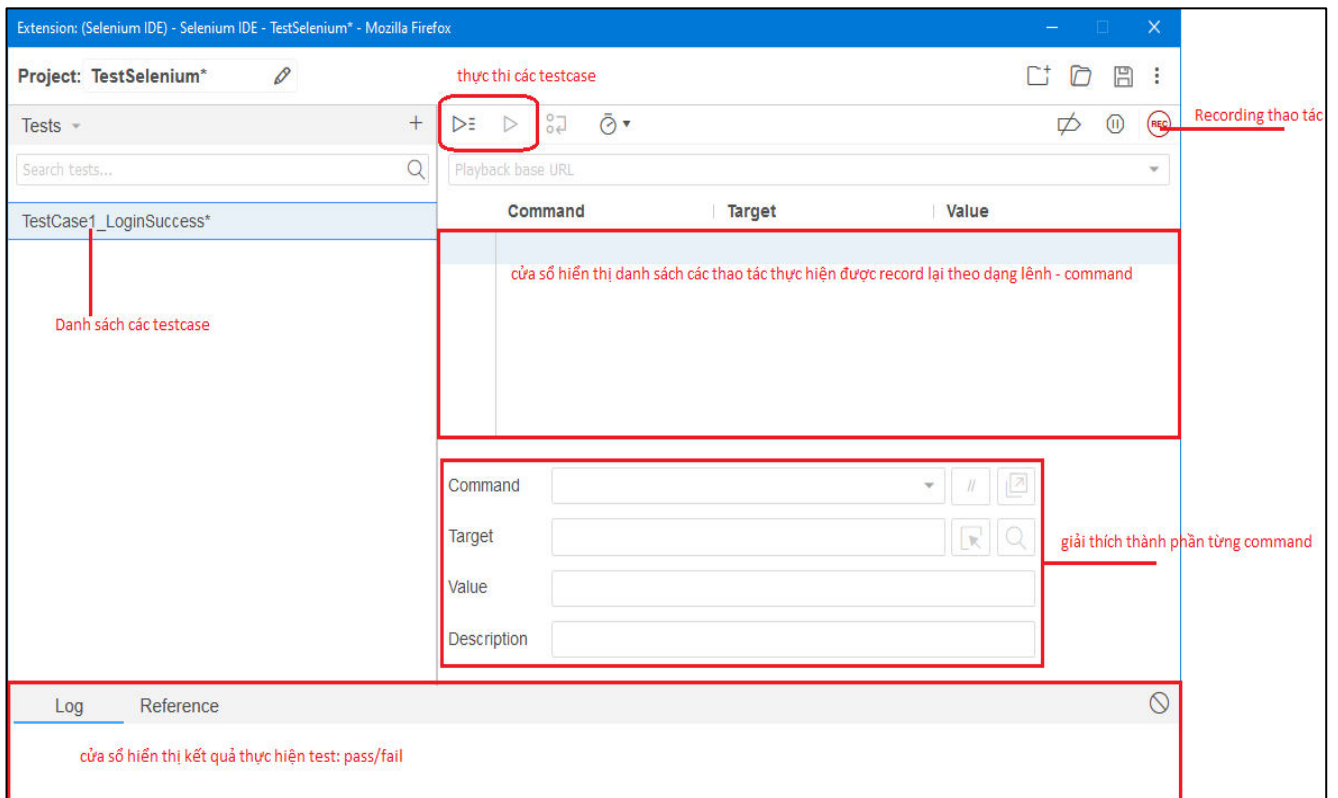


##### Bước 2: Giao diện làm việc của Selenium

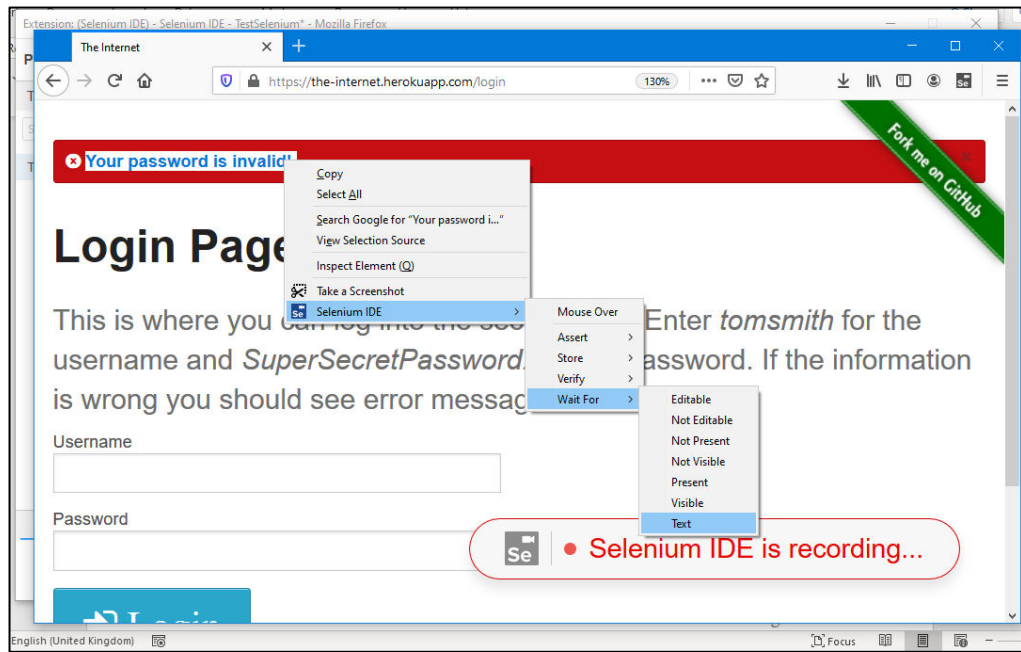
- Màn hình khởi động ban đầu, tạo project mới:



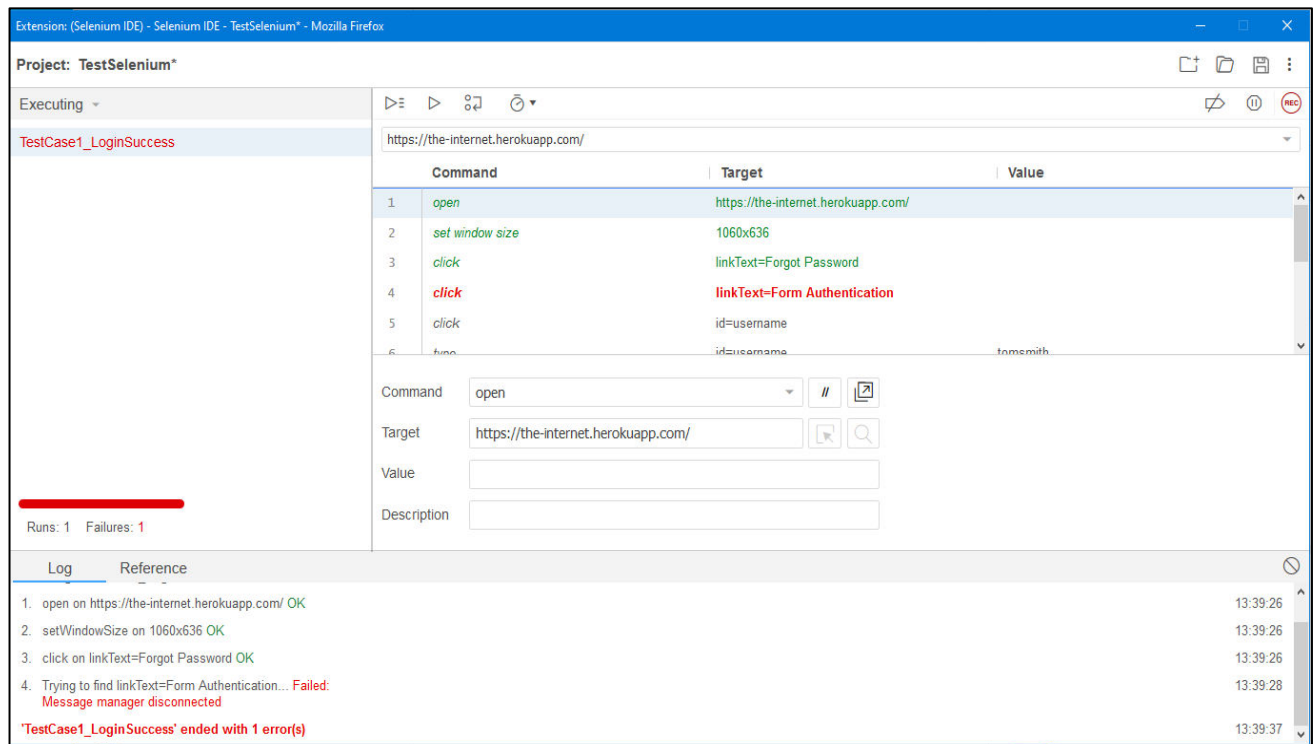
## - Màn hình làm việc của Selenium



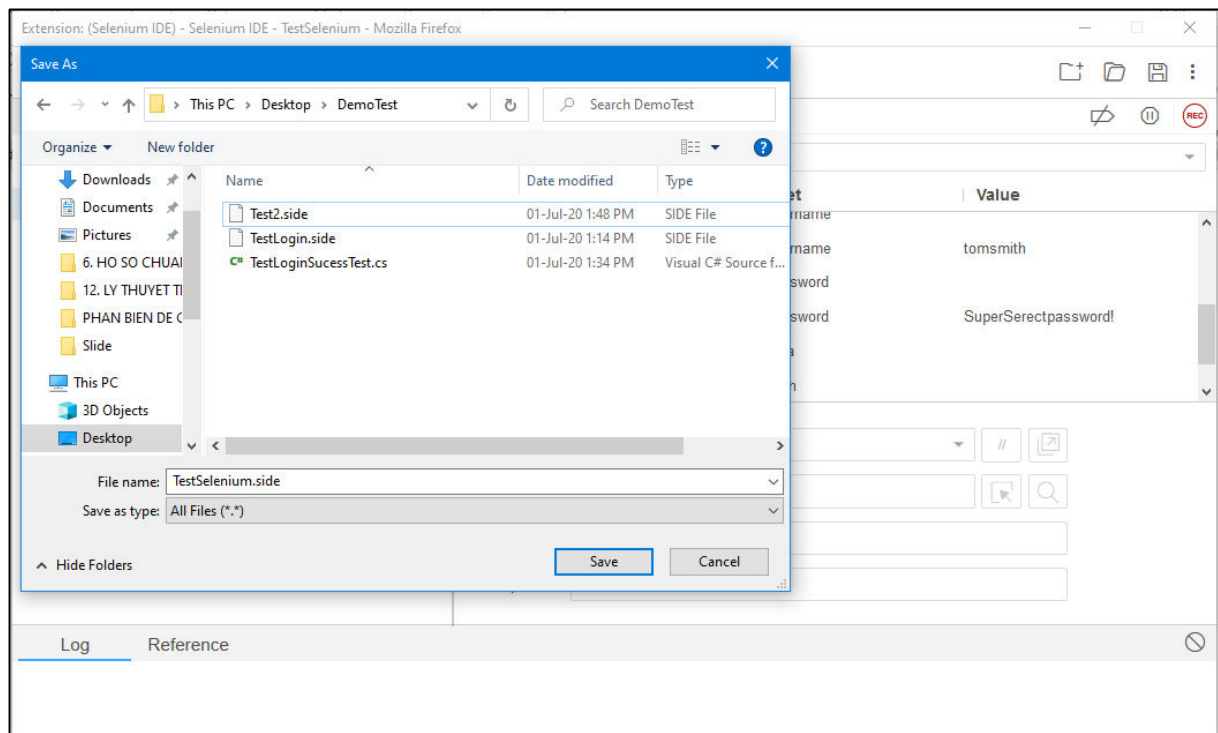
## Bước 3. Record and Play Back, chèn assert để bắt các thao tác cần check lỗi



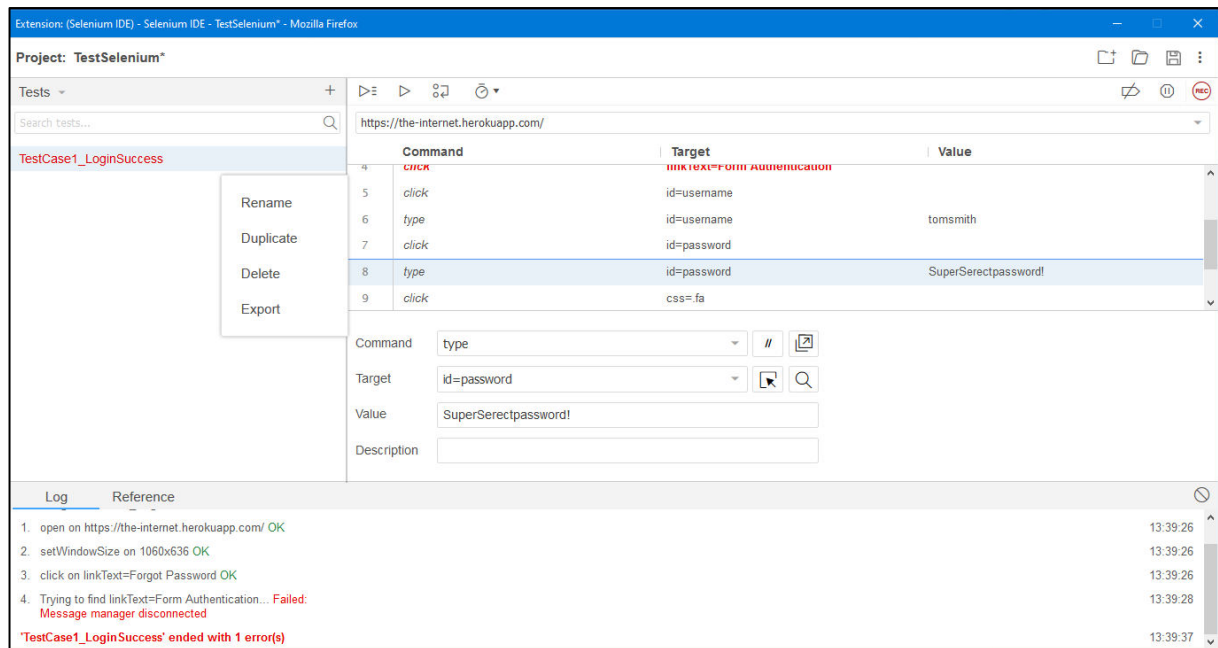
- Chạy testcase: thao tác nào pass có màu xanh lá, thao tác nào fail có màu đỏ.



#### Bước 4: Lưu file recording của Selenium dạng \*.side



## Bước 5: Export thành Script dạng NUnit



Select language

☒ C# NUnit

☐ C# xUnit

☐ Java JUnit

☐ JavaScript Mocha

☐ Python pytest

☐ Ruby RSpec

☐ Include origin tracing code comments

☐ Include step description as a separate comment

☐ Export for use on Selenium Grid

EXPORT

CANCEL

- Ví dụ: file NUnit được export từ Selenium

```
// Generated by Selenium IDE
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.Remote;
using OpenQA.Selenium.Support.UI;
using OpenQA.Selenium.Interactions;
using NUnit.Framework;
using System.Collections.Generic;

[TestFixture]
public class TestLoginSucessTest {
    private IWebDriver driver;
    public IDictionary<string, object> vars {get; private set;}
    private IJavaScriptExecutor js;

    List<String> us = new List<String>(new String[] { "tomsmith", "ngan" });
    List<String> pw = new List<String>(new String[] { "SuperSecretPassword!", "111" });

    [SetUp]
    public void SetUp() {
        driver = new FirefoxDriver();
        js = (IJavaScriptExecutor)driver;
        vars = new Dictionary<string, object>();
    }
    [TearDown]
    protected void TearDown() {
        driver.Quit();
    }
}
```

```

[Test]
public void testLoginSucess() {
    // Test name: TestLoginSucess

    // Step # | name | target | value
    // 1 | open | https://the-internet.herokuapp.com/ |
    driver.Navigate().GoToUrl("https://the-internet.herokuapp.com/");
    // 2 | setWindowSize | 1207x680 |
    driver.Manage().Window.Size = new System.Drawing.Size(1207, 680);
    // 3 | click | linkText=Form Authentication |
    driver.FindElement(By.LinkText("Form Authentication")).Click();
    // 4 | click | id=username |
    driver.FindElement(By.Id("username")).Click();
    // 5 | type | id=username | tomsmith
    // driver.FindElement(By.Id("username")).SendKeys("tomsmith");
    driver.FindElement(By.Id("username")).SendKeys("tomsmith");
    // 6 | click | css=.row:nth-child(2) > .large-6 > label |
    driver.FindElement(By.CssSelector(".row:nth-child(2) > .large-6 > label")).Click();
    // 7 | type | id=password | SuperSecretPassword!
    driver.FindElement(By.Id("password")).SendKeys("SuperSecretPassword!");
    // 8 | click | css=.fa |
    driver.FindElement(By.CssSelector(".fa")).Click();
    // 9 | assertText | id=flash | Your password is invalid!\n
    Assert.That(driver.FindElement(By.Id("flash")).Text, Is.EqualTo("Your password is
invalid!\\n\\n"));
}
}

```

## 2. Một số lệnh thông dụng của Selenium

Command	Mô tả
open	Mở ra một đường link cụ thể trên trình duyệt.
assertTitle, VerifyTitle	Trả về title của trang web hiện tại và so sánh nó với 1 title cụ thể.
assertElementPresent, verifyElementPresent	Xác minh/ xác nhận sự xuất hiện của 1 phần tử trên web.
assertTextPresent, verifyTextPresent	Xác minh/ xác nhận các text hiện có trên 1 trang web.
type, typeKeys, sendKeys	Nhập 1 giá trị (dạng String) vào 1 phần tử cụ thể trên web.
Click, clickAt, clickAndWait	Kích vào 1 phần tử web cụ thể trên trang.
waitForPageToLoad	Tạm dừng việc thực hiện cho đến khi trang load thành công.

waitForElement Present	Tạm dừng thực hiện cho đến khi hiển thị 1 phần tử cụ thể.
chooseOkOnNext, Confirmation, chooseCancelOn, NextConfirmation	Kích vào button "OK" hoặc "Cancel" khi hộp xác nhận tiếp theo hiển thị.

## II. Phần thực hành có hướng dẫn

**Kịch bản test chức năng đăng nhập trên website: <https://id.zing.vn>**

Toàn bộ quá trình tạo ra kịch bản Selenium IDE được chia làm 3 phần:

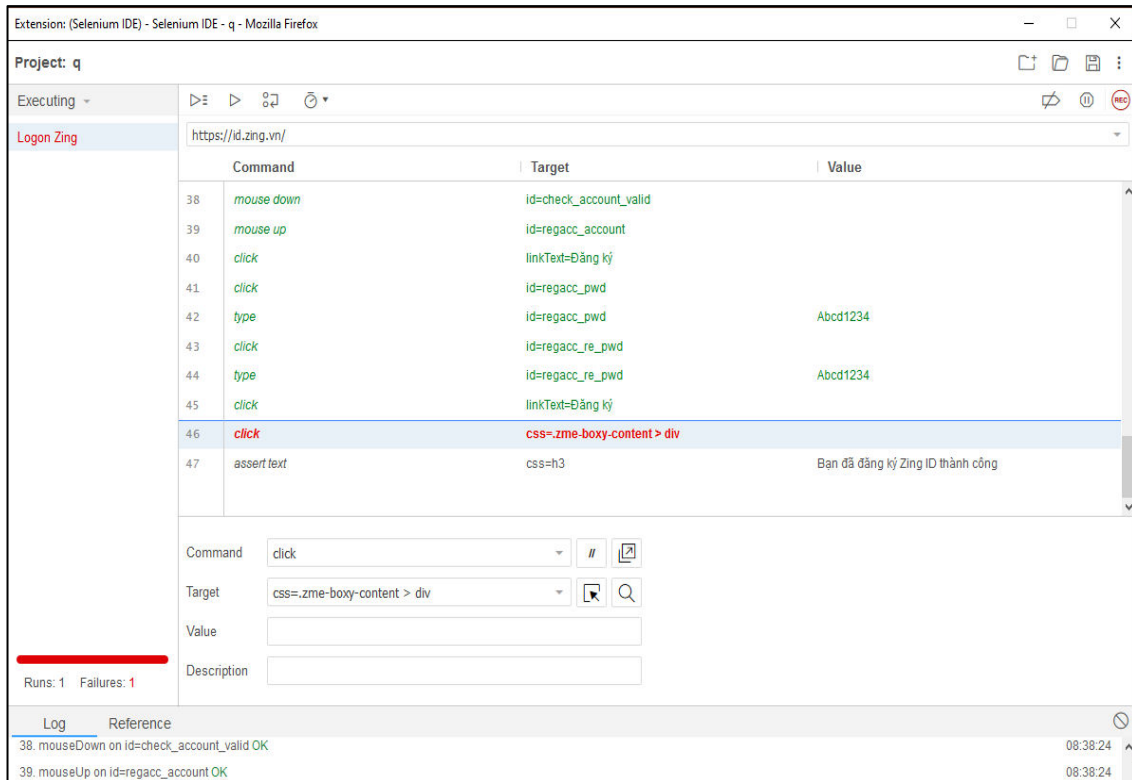
- ❖ *Phần 1: Recording* - Selenium IDE hỗ trợ người dùng ghi lại các tương tác của người dùng với trình duyệt. Các hoạt động được ghi lại đó tạo nên kịch bản lệnh Selenium IDE.
  - Mở trang web: <http://id.zing.vn/>
  - Check title của website.
  - Thực hiện đăng ký tài khoản.
  - Xác minh login thành công.
- ❖ *Phần 2: Playing back* - Trong giai đoạn này, chúng ta sẽ thực thi các kịch bản đã được ghi lại để xác minh và kiểm soát độ ổn định và tỷ lệ thành công của kịch bản.

### ***Kịch bản đăng nhập Zing:***

- Bước 1: Khởi động Firefox và chạy Selenium IDE.
- Bước 2: Nhập địa chỉ website vào thanh Url "http://id.zing.vn/". Khi mở Selenium, mặc định nút Record đang được bật.
- Bước 3: Click "Đăng ký tài khoản" để mở màn hình nhập thông tin người dùng.
- Bước 4: Xác minh title của website mở đã chính xác chưa bằng cách kích chuột phải trên màn hình trang hiện tại, chọn Selenium IED → Assert/Verify... để chọn các điểm chèn test.
- Bước 5: Tiến hành thực hiện record các thao tác bằng Selenium. Lưu ý các điểm cần check:
  - ❖ Tên username đã tồn tại.
  - ❖ Thông báo yêu cầu điền nội dung và username và password.
  - ❖ Thông báo yêu cầu cách đặt password
  - ❖ Check thông báo tạo tài khoản thành công.
- Tạo bộ test data để đăng nhập
  - + Họ tên account
  - + user name account

- + password account
- + giới tính và ngày sinh

Ví dụ : kết quả sau khi record kịch bản tạo id\_zing



❖ **Phần 3: Saving** - Khi ghi được một kịch bản ổn định, chúng ta có thể lưu lại để chạy hoặc test hồi quy trong tương lai. Lưu ý: để chạy lại testcase này cần đổi bộ testdata cho phù hợp, tương ứng account mới, không tồn tại

Kết quả export về NUnit C#:

```
[TearDown]
protected void TearDown() {
    driver.Quit();
}

[Test]
public void logonZing() {
    driver.Navigate().GoToUrl("https://id.zing.vn/");
    driver.Manage().Window.Size = new System.Drawing.Size(1207, 728);
    driver.FindElement(By.LinkText("Đăng ký ngay")).Click();
    driver.FindElement(By.Id("regacc_fullname")).Click();
    driver.FindElement(By.Id("regacc_fullname")).SendKeys("Nguyen Thanh Hoa");
    driver.FindElement(By.Id("regacc_account")).Click();
    driver.FindElement(By.Id("regacc_account")).SendKeys("thanhhoa");
    driver.FindElement(By.Id("regacc_pwd")).Click();
    driver.FindElement(By.Id("regacc_pwd")).SendKeys("Abc1111");
    driver.FindElement(By.Id("regacc_re_pwd")).Click();
    driver.FindElement(By.Id("regacc_re_pwd")).SendKeys("Abc1111");
    driver.FindElement(By.Name("gender")).Click();
    driver.FindElement(By.LinkText("Đăng ký")).Click();
    Assert.That(driver.FindElement(By.CssSelector("#regacc_pwd_error .tipcontent")).Text, Is.EqualTo("Vui lòng nhập mật khẩu dài 6-32 ký tự, có ký tự chữ số, chữ hoa và chữ thường"));
    driver.FindElement(By.Id("regacc_pwd")).Click();
    driver.FindElement(By.Id("regacc_pwd")).SendKeys("Abc1234");
    driver.FindElement(By.Id("regacc_re_pwd")).Click();
    driver.FindElement(By.Id("regacc_re_pwd")).SendKeys("Abc1234");
    driver.FindElement(By.LinkText("Đăng ký")).Click();
    driver.FindElement(By.Id("dob")).Click();
    {
        var dropdown = driver.FindElement(By.Id("dob"));
        dropdown.FindElement(By.XPath("//option[. = '2']")).Click();
    }
}
```



### III. Phần thực hành trên lớp

**Bài 1.** Xây dựng kịch bản và thực hiện test chức năng **đăng nhập thành công 1 tài khoản/ đăng nhập 1 tài khoản không tồn tại** đã có trong website : <https://id.zing.vn>

Thực hiện kiểm tra: các vị trí từ 1 → 8 theo hình mẫu.

The screenshot shows the Zing ID login interface. Annotations are as follows:

- 1: User name "Chào David Smith" in the top right header.
- 2: "THOAT" (Logout) button in the top right header.
- 3: Search bar in the top navigation bar.
- 4: Account ID "davidsmith1112" and account number "Mã số tài khoản: 9600780377" in the top navigation bar.
- 5: Username field containing "davidsmith1112".
- 6: Email field containing "(Chưa có thông tin)".
- 7: Phone number field containing "(Chưa có thông tin)".
- 8: Password field containing "\*\*\*\*\*".

- Record và playback, thay đổi testdata và test.
- Export thành file Nunit, chỉnh sửa theo bộ test data mới.

**Bài 2.** Xây dựng kịch bản và thực hiện test chức năng cập nhật email, số điện thoại và mật khẩu trên tài khoản id\_Zing. Sau đó cập nhật lại kịch bản test bài 1, có check thông tin email và sdt đã cập nhật.

- Record và playback, thay đổi testdata và test.
- Export thành file Nunit, chỉnh sửa theo bộ test data mới.

**Bài 3.** Truy cập vào website demo của Selenium <https://parabank.parasoft.com/>

- Xây dựng kịch bản, thực hiện Record và playback, thay đổi testdata và test. Export thành file Nunit, chỉnh sửa theo bộ test data mới cho các yêu cầu của các chức năng:
  1. Tạo mới tài khoản ngân hàng, kiểm tra tên tài khoản mới có xuất hiện trên website tại vị trí sau chữ welcome.
  2. Chuyển khoản từ tài khoản A sang tài khoản B. Kiểm tra thông tin sau khi chuyển.
  3. Thanh toán hoá đơn (Bill bay). Kiểm tra thông tin sau khi thanh toán.

**Bài 4.** Truy cập vào website demo của Selenium <https://magento.softwaretestingboard.com/>

- Xây dựng kịch bản, thực hiện Record và playback, thay đổi testdata và test. Export thành file Nunit, chỉnh sửa theo bộ test data mới cho các yêu cầu của các chức năng:
  1. Mua một sản phẩm và thanh toán.

2. Mua nhiều hơn 2 sản phẩm và thanh toán.
3. Chọn nhiều hơn 3 sản phẩm, xóa sản phẩm trong giỏ hàng và thanh toán.

#### **IV. Phần thực hành về nhà.**

**Bài 5.** Truy cập vào website demo của Selenium <https://the-internet.herokuapp.com> thực hiện các yêu cầu sau: xây dựng kịch bản test chức năng cho các trang sau

1/ [https://the-internet.herokuapp.com/key\\_presses](https://the-internet.herokuapp.com/key_presses) : test gõ phím

2/ <https://the-internet.herokuapp.com/inputs>: input number

#### **Bài 6.**

Xây dựng kịch bản và thực hiện test các chức năng mua hàng trực tuyến trên website demo <http://automationpractice.com>

#### **Yêu cầu:**

- 1/ Thực hiện chọn ít nhất 2 sản phẩm, bỏ vào giỏ hàng và tiến hành thanh toán với người dùng chưa có tài khoản.
- 2/ Thực hiện tạo tài khoản.
- 3/ Đăng nhập tài khoản thành công
- 4/ Thực hiện mua hàng, có các bước chọn và bỏ chọn sản phẩm, check việc thay đổi số lượng và đơn giá sản phẩm.
- 5/ Thực hiện tìm kiếm và so sánh kết quả tìm kiếm sản phẩm.

**--HẾT--**