

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO BÀI TẬP LỚN
MÔN: KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

TÌM HIỂU CÔNG CỤ KIỂM THỬ SILENIUM

Giảng viên: Nguyễn Thu Trang

Sinh viên

Họ và tên	MSSV	Lớp
Nguyễn Văn Huy	18020651	INT3117 1
Vũ Thị Oanh	18020988	INT3117 2

Hà Nội - 2021

MỤC LỤC

MỤC LỤC	2
I. Tổng quan về Selenium	4
Giới thiệu chung	4
Sự ra đời và phát triển	4
Khả năng hỗ trợ	4
Đặc điểm nổi bật	5
II. Các công cụ thành phần	6
Selenium IDE (Môi trường phát triển tích hợp Selenium)	6
Giới thiệu chung	6
Khả năng hỗ trợ	6
Ưu điểm	7
Nhược điểm	7
Selenium RC (Selenium Remote Control)	7
Giới thiệu chung	7
Thành phần của Selenium RC	8
Ưu điểm	9
Nhược điểm	9
Webdriver	9
Giới thiệu chung	9
Cách thức hoạt động	10
Ưu điểm	11
Nhược điểm	11
Selenium Grid	12
Giới thiệu chung	12

Thành phần của Selenium Grid	12
Ưu điểm	13
III. Cách cài đặt, yêu cầu cài đặt	13
Selenium IDE	13
Selenium Webdriver	16
IV. Giới thiệu dự án open source/ bài tập lớn	18
V. Báo cáo kết quả phân tích kiểm thử	21
Kiểm thử chức năng Đăng ký tài khoản	21
Kiểm thử chức năng Đăng nhập	23
Kiểm thử chức năng Tạo bài viết	25
Kiểm thử chức năng Bình luận bài viết	27
VI. So sánh - Tổng kết	29
VII. Nguồn tham khảo	31

I. Tổng quan về Selenium

1. Giới thiệu chung

Selenium là một công cụ kiểm thử phần mềm tự động mã nguồn mở dành cho các ứng dụng web trên nhiều trình duyệt (như Google Chrome, IE, Safari,...) và hệ điều hành khác nhau (như Window, Mac OS, Linux, ...).

Selenium được sử dụng để có thể tự động hóa cho các thao tác với trình duyệt, nói một cách đơn giản hơn, Selenium hỗ trợ giả lập nhằm tăng khả năng tương tác trên trình duyệt.

Ngoài mục đích sử dụng trong kiểm thử, Selenium có thể được sử dụng khi xây dựng project để tự động hóa các công việc phải lặp đi lặp lại nhiều lần.

2. Sự ra đời và phát triển

Selenium được ra đời vào năm 2004 bởi Jason Huggins - một kỹ sư tại ThoughtWorks.

Khi làm việc trên một ứng dụng web yêu cầu kiểm thử thường xuyên, ông nhận ra rằng lặp đi lặp lại việc kiểm thử thủ công không còn đem lại hiệu quả. Nhằm tăng hiệu quả kiểm thử bằng cách kiểm soát các hành động của trình duyệt một cách tự động, ông đã tạo ra chương trình “JavaScriptTestRunner”.

Nhìn thấy được tiềm năng của chương trình trong việc kiểm thử tự động các ứng dụng web khác, Jason Huggins đã tạo ra JavaScriptRunner, sau được đổi tên thành Selenium.

Năm 2008, Webdriver và Selenium RC (2 thành phần của Selenium) được hợp nhất tạo thành một công cụ mạnh mẽ hơn là Selenium 2 và dần được phát triển thành Selenium 3, 4.

3. Khả năng hỗ trợ

Selenium là bộ công cụ được sử dụng phổ biến nhất trong việc kiểm thử web tự động. Selenium có khả năng:

- Selenium giúp người dùng thực hiện được kiểm thử hồi quy một cách dễ dàng và thường xuyên hơn.

- Đưa ra những phản hồi nhanh chóng và kịp thời tới người dùng.
- Không giới hạn việc thực hiện lặp lại các công việc trong quá trình kiểm thử.
- Cung cấp các tài liệu kiểm thử nhanh chóng và chính xác, đồng thời có thể giúp người dùng tùy chỉnh được các nhược điểm của báo cáo.
- Tìm ra được những nhược điểm bị bỏ qua trong quá trình kiểm thử bằng phương pháp thủ công.

4. Đặc điểm nổi bật

- Selenium là một bộ công cụ kiểm thử tự động mã nguồn mở dành cho các ứng dụng web trên nhiều trình duyệt và hệ điều hành khác nhau.
- Cộng đồng hỗ trợ mạnh mẽ, thường xuyên được phát triển và cải tiến mạnh mẽ.
- Selenium hỗ trợ viết ca kiểm thử với nhiều ngôn ngữ lập trình khác nhau, như Python, Java, .Net,... Đồng thời, với mỗi ngôn ngữ lập trình, cũng sẽ có nhiều framework hỗ trợ như Behave, Lettuce, ... (Python); Gauge, JBehave, ... (Java); CucumberJS, Intern, ... (JavaScript); ...
- Hỗ trợ trên nhiều hệ điều hành khác nhau mà gần như không cần chỉnh sửa script.
- Có thể chạy ca kiểm thử tại background, giúp tiết kiệm tài nguyên máy.
- Chỉ hỗ trợ tương tác với trình duyệt web mà không hỗ trợ với ứng dụng di động.

II. Các công cụ thành phần

Selenium là một bộ công cụ kiểm thử, mỗi bộ phận phục vụ cho một nhu cầu kiểm thử khác nhau. Selenium gồm bốn thành phần:

- Selenium IDE (Môi trường phát triển tích hợp Selenium)
- Selenium RC (Selenium Remote Control)
- Webdriver
- Selenium Grid

1. Selenium IDE (Môi trường phát triển tích hợp Selenium)

a. Giới thiệu chung

Selenium IDE, được tạo ra vào năm 2006, là một plug-in trên Firefox, có khả năng tự động hóa trình duyệt thông qua tính năng ghi hoặc thực hiện các thao tác dựa trên một quy trình hoặc một ca kiểm thử. Selenium IDE giúp tăng thêm tốc độ trong việc tạo ra các ca kiểm thử.

Đây là framework đơn giản nhất trong bộ Selenium, có thể dễ dàng cài đặt. Tuy nhiên, chính vì tính đơn giản nên Selenium IDE chỉ có thể sử dụng làm công cụ tạo mẫu.

b. Khả năng hỗ trợ

- Kiểm thử hồi quy: Selenium ghi lại nhiều bộ định vị cho mỗi phần tử mà nó tương tác. Nếu có một bộ định vị không thành công khi phát lại, các bộ định vị khác sẽ được thử cho đến khi có một bộ định vị thành công.
- Tái sử dụng ca kiểm thử: Thông qua việc sử dụng lệnh run, người dùng có thể sử dụng lại một ca kiểm thử bên trong một ca kiểm thử khác.
- Luồng điều khiển: Selenium IDE đi kèm với các lệnh cho phép người dùng thêm logic và vòng lặp có điều kiện vào các ca kiểm thử. Điều này cho phép người dùng thực thi các lệnh hoặc một tập hợp các lệnh khi đáp ứng điều kiện nhất định trong ứng dụng, hoặc

cho phép thực thi các lệnh lặp đi lặp lại dựa trên một điều kiện xác định.

- Khả năng gỡ lỗi: Cho phép gỡ lỗi một cách dễ dàng bằng cách thiết lập các điểm ngắt / tạm dừng trên các ca kiểm thử.

c. Ưu điểm

- Đơn giản, dễ dàng cài đặt.
- Dễ dàng sử dụng, không yêu cầu kinh nghiệm lập trình cao, chỉ cần có chút kiến thức về HTML và DOM là có thể sử dụng.
- Có thể xuất các ca kiểm thử đã được tạo để sử dụng trong Selenium RC và Webdriver.
- Cung cấp chức năng báo cáo kết quả hoặc hỗ trợ cần thiết khi sử dụng.
- Có khả năng tích hợp cùng các extension khác nhau.

d. Nhược điểm

- Chỉ có thể cài đặt trên trình duyệt Firefox.
- Chỉ được sử dụng làm công cụ tạo mẫu kiểm thử.
- Không có khả năng thực hiện các phép toán, câu lệnh phức tạp hoặc câu lệnh có điều kiện.
- Hiệu năng hoạt động thấp hơn so với Selenium RC và Webdriver.

2. Selenium RC (Selenium Remote Control)

a. Giới thiệu chung

Selenium RC được tạo ra nhằm giải quyết vấn đề người dùng phải cài đặt toàn bộ ứng dụng kiểm thử và máy chủ web trên máy tính cục bộ.

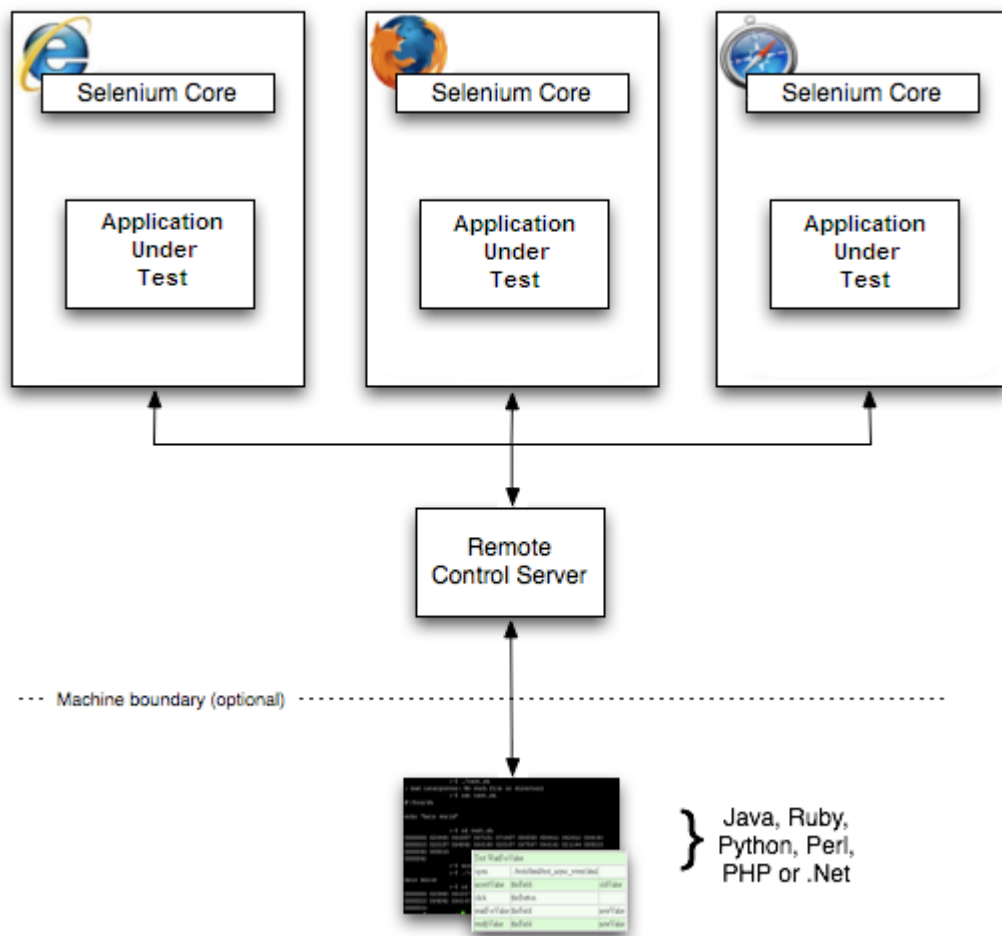
Đây là một framework kiểm thử hàng đầu của bộ công cụ Selenium, là công cụ kiểm thử web tự động đầu tiên cho phép người dùng sử dụng nhiều loại ngôn ngữ lập trình như C#, PHP, Python, Ruby,....

b. Thành phần của Selenium RC

Selenium RC bao gồm:

- Selenium server (máy chủ Selenium): máy chủ Selenium khởi chạy và tương tác với trình duyệt web.
- Các thư viện máy khách cung cấp giao diện giữa ngôn ngữ lập trình và máy chủ Selenium.

Windows, Linux, or Mac (as appropriate)...



Hình 1. Sơ đồ kiến trúc của Selenium RC

Cách hoạt động: Các thư viện máy khách giao tiếp với máy chủ thông qua mỗi lệnh Selenium để thực thi. Sau đó máy chủ chuyển lệnh Selenium đến trình duyệt bằng các lệnh JS Selenium-Core. Trình duyệt sẽ thực hiện các lệnh Selenium.

c. Ưu điểm

- Có thể chạy trên nhiều trình duyệt (kể cả với các trình duyệt mới) và hệ điều hành khác nhau.
- Hỗ trợ các hoạt động lặp lại hoặc có điều kiện.
- Có API hoàn thiện, có thể hỗ trợ kiểm thử data-driven.
- Hiệu suất thực thi nhanh hơn Selenium IDE.

d. Nhược điểm

- Cài đặt phức tạp hơn Selenium IDE.
- Yêu cầu người dùng cần có kiến thức về lập trình.
- Yêu cầu máy chủ Selenium RC để thực thi.
- Sự tương tác với trình duyệt ít thực tế. Quá trình tương tác với trình duyệt cần thông qua JavaScript.
- Thời gian thực thi chậm hơn Webdriver.

3. Webdriver

a. Giới thiệu chung

Webdriver được tạo ra vào khoảng năm 2006, khi các trình duyệt và ứng dụng web ngày càng trở nên mạnh mẽ và hạn chế hơn đối với các chương trình JavaScript.

Đây là framework kiểm thử đa nền tảng đầu tiên có thể kiểm soát trình duyệt từ cấp hệ điều hành. Webdriver thực hiện mọi tự động hóa tương tác với các trình duyệt nhằm hướng tiếp cận hiện đại và ổn định hơn. Quá trình tương tác với trình duyệt đều sẽ được gửi trực tiếp từ Selenium Webdriver mà không cần thông qua JavaScript như Selenium RC.

Webdriver hỗ trợ các ngôn ngữ lập trình giống với Selenium RC.

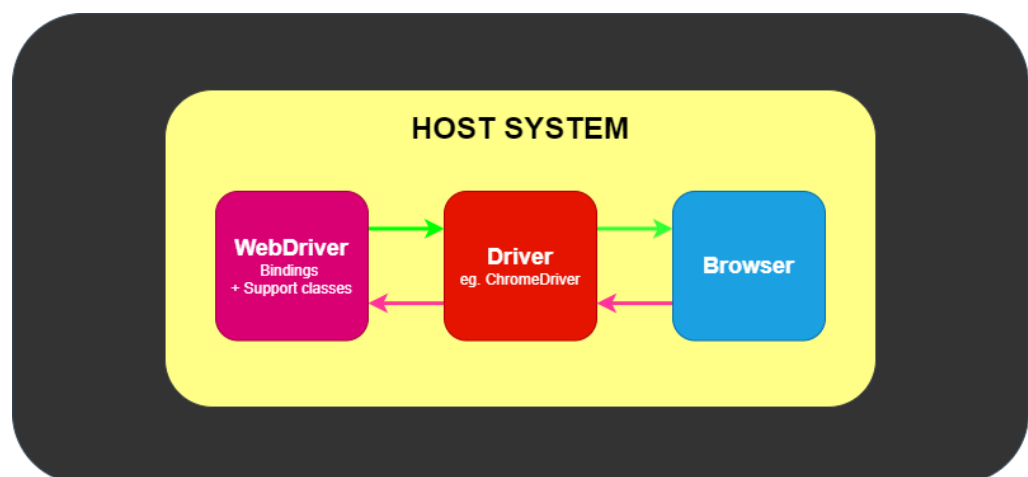
b. Cách thức hoạt động

Selenium hỗ trợ tự động hóa tất cả các trình duyệt lớn hiện nay nhờ vào việc sử dụng Webdriver. Webdriver là API và giao thức cho phép điều khiển các hành vi của trình duyệt. Mỗi trình duyệt được hỗ trợ bởi bộ Webdriver cụ thể, được gọi là trình điều khiển. Trình điều khiển là thành phần chịu trách nhiệm ủy quyền cho trình duyệt và xử lý các giao tiếp giữa Selenium và trình duyệt.

Giao tiếp giữa Webdriver và trình duyệt là giao tiếp hai chiều: Webdriver truyền các câu lệnh tới trình duyệt thông qua trình điều khiển và nhận lại thông tin ngay trong quá trình đó.

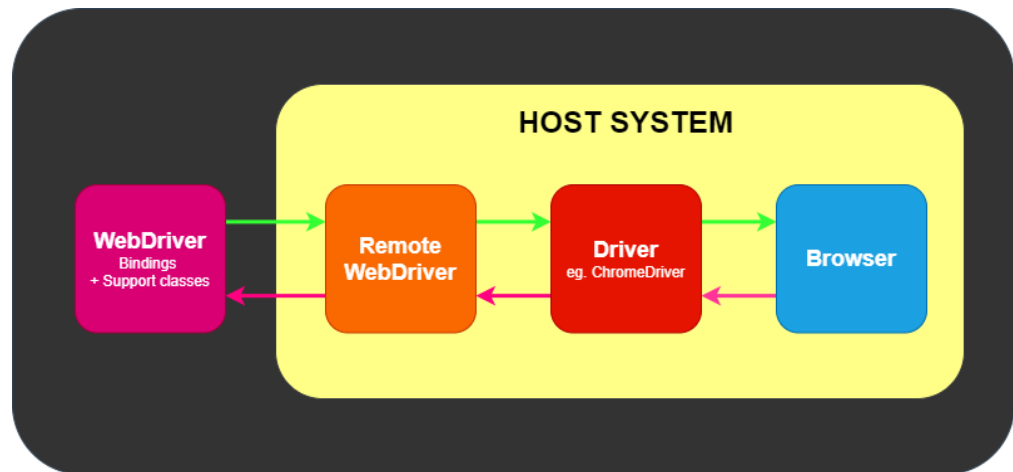
Có 2 cách để Webdriver giao tiếp với trình duyệt:

- Giao tiếp trực tiếp: Webdriver giao tiếp trực tiếp với trình duyệt thông qua trình điều khiển.



Hình 2. Sơ đồ giao tiếp trực tiếp của Webdriver

- Giao tiếp từ xa: Webdriver giao tiếp với trình duyệt thông qua máy chủ Selenium hoặc RemoteWebDriver. RemoteWebDriver chạy trên cùng hệ thống với trình điều khiển và trình duyệt.



Hình 3. Sơ đồ giao tiếp từ xa của Webdriver bằng RemoteWebDriver

c. Ưu điểm

- Cài đặt đơn giản hơn Selenium RC do không yêu cầu các thành phần khác (máy chủ Selenium).
- Có thể thực hiện giao tiếp trực tiếp với trình duyệt.
- Quá trình tương tác với trình duyệt tương tự như thao tác của một người dùng thật.
- Thao tác dễ dàng với nhiều phép toán logic hoặc có điều kiện phức tạp.
- Tốc độ xử lý nhanh hơn Selenium RC và Selenium IDE.

d. Nhược điểm

- Yêu cầu người dùng có kiến thức về lập trình.
- Không sẵn sàng hỗ trợ đối với các trình duyệt mới mà cần phải thiết lập trình điều khiển cho trình duyệt tương ứng.
- Không có cơ chế built-in cho việc ghi lại thông điệp trong thời gian chạy và sinh kết quả kiểm thử.

4. Selenium Grid

a. Giới thiệu chung

Selenium Grid được tạo ra nhằm giải quyết nhu cầu giảm thiểu thời gian kiểm tra càng nhiều càng tốt. Selenium Grid là một công cụ được sử dụng cùng với Selenium RC để chạy kiểm thử song song trên các máy khác nhau và các trình duyệt khác nhau cùng một thời điểm.

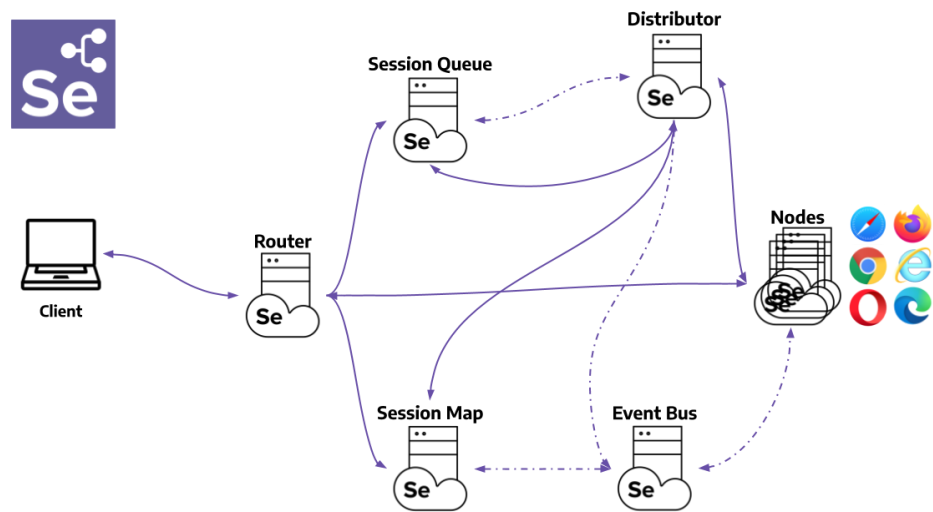
Có 2 lý do chính để cân nhắc việc sử dụng Selenium Grid là:

- Các kịch bản kiểm thử yêu cầu phải được kiểm thử trên nhiều trình duyệt, hệ điều hành khác nhau trong cùng một thời điểm. Đảm bảo ứng dụng đang kiểm thử hoàn toàn tương thích với nhiều loại trình duyệt, hệ điều hành.
- Để tiết kiệm thời gian thực thi bộ ca kiểm thử. Thay vì thực thi 1 ca kiểm thử 4 lần cho 4 trình duyệt khác nhau thì Selenium Grid có thể chạy 4 ca kiểm cùng một lúc cho 4 trình duyệt khác nhau.

b. Thành phần của Selenium Grid

Selenium Grid gồm 2 thành phần chính:

- Hub: đóng vai trò như một máy chủ, là nơi gửi lệnh điều khiển đến các mô hình khác trong mô hình thực thi kiểm thử. Hub chỉ có thể được cài đặt duy nhất trên một máy tính.
- Nodes: là các thể hiện được kết nối vào Hub để thực thi các kịch bản kiểm thử. Một mô hình Grid có thể có nhiều nodes. Các nodes có thể được cài đặt trên nhiều máy tính với nhiều trình duyệt khác nhau.



Hình 4. Sơ đồ mô hình đầy đủ của Selenium Grid

Cách hoạt động: Từ Hub, thực hiện tạo các Nodes. Các nodes chính là nơi thực thi các ca kiểm thử. Còn Hub có vai trò tìm các node có điều kiện tương ứng với yêu cầu đầu vào và chuyển các ca kiểm thử đến đúng môi trường (trình duyệt và hệ điều hành) cần thực thi.

c. Ưu điểm

- Cho phép chạy đồng thời các kiểm thử trên nhiều trình duyệt và môi trường (máy tính) khác nhau.
- Tiết kiệm thời gian kiểm thử.

III. Cách cài đặt, yêu cầu cài đặt

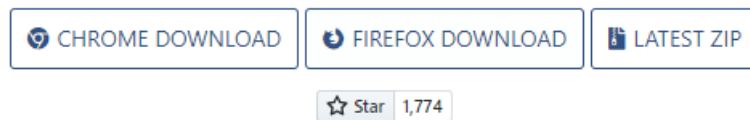
1. Selenium IDE

Bước 1: Tải Selenium IDE

Truy cập đường dẫn <https://www.selenium.dev/selenium-ide/> và chọn phiên bản tương ứng với trình duyệt hiện tại đang sử dụng:

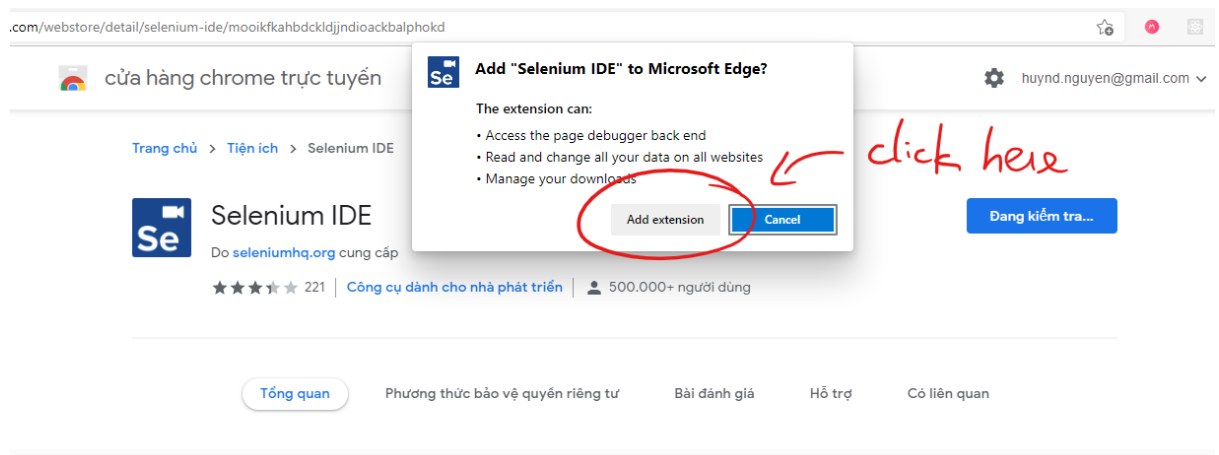
Selenium IDE

Open source record and playback test automation for the web



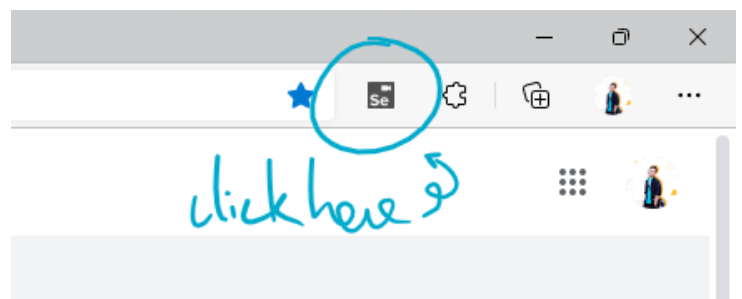
Hình 5. Hướng dẫn cài đặt Selenium IDE

Ví dụ, nếu sử dụng trình duyệt nhân Chromium, ta tiến hành cài đặt từ Chrome Store.

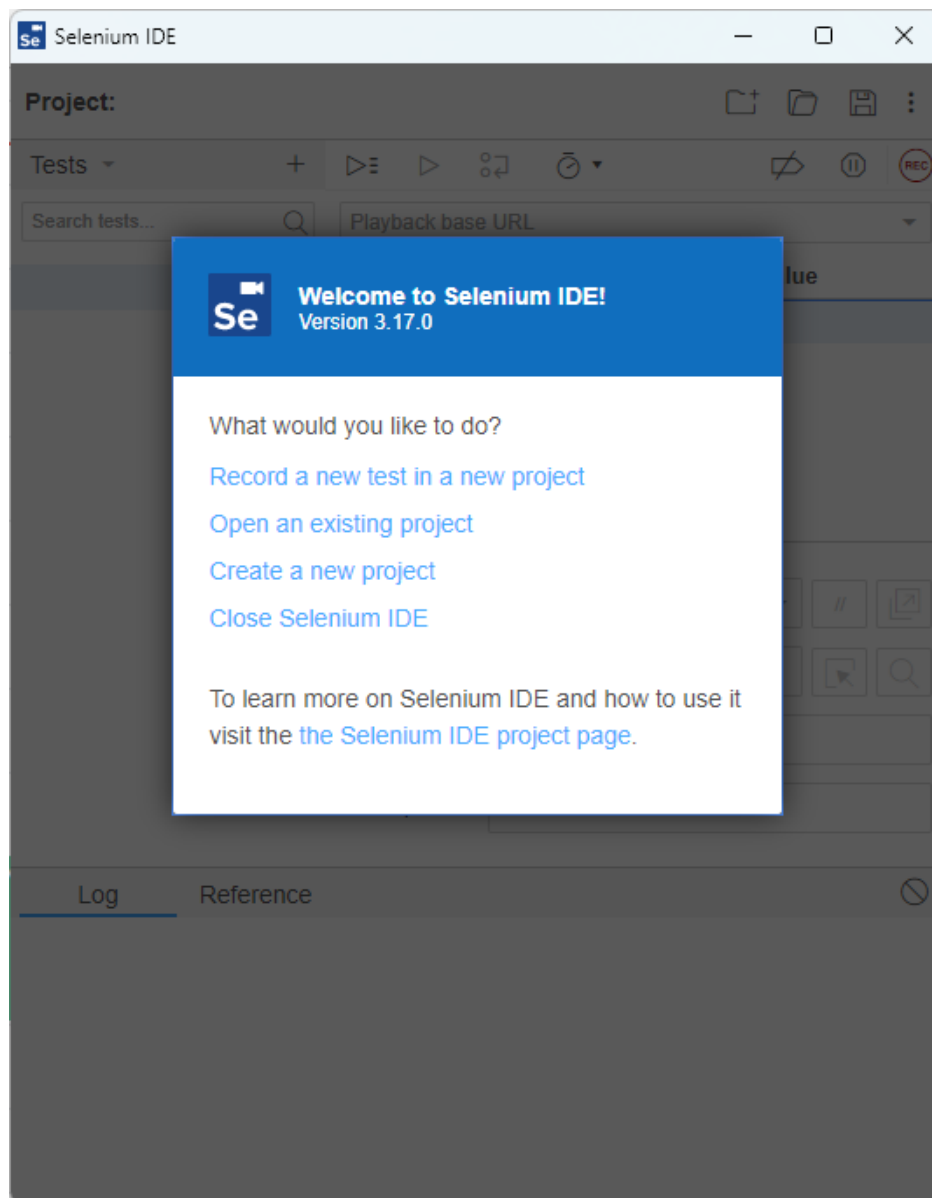


Hình 6. Hướng dẫn cài đặt Selenium IDE

Sau khi cài đặt thành công, ta ấn vào biểu tượng extension để khởi động Selenium



Hình 7. Hướng dẫn cài đặt Selenium IDE



Hình 8. Hướng dẫn cài đặt Selenium IDE

2. Selenium Webdriver

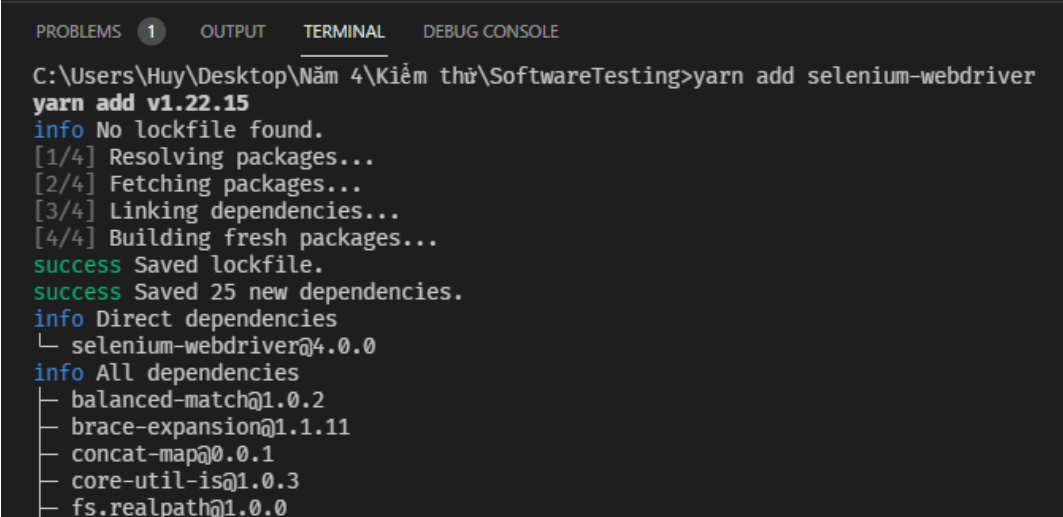
Thiết lập Selenium khá khác so với thiết lập các công cụ trả phí khác. Để sử dụng Selenium trong dự án, cần cài đặt các thư viện dành cho ngôn ngữ tương ứng. Ngoài ra, cần cài đặt Webdriver cho các trình duyệt mà bạn muốn tự động hóa và chạy kiểm thử trên đó.

Cài đặt Selenium Webdriver có thể được chia ra làm 3 bước chính:

Bước 1. Cài đặt thư viện Selenium Webdriver cho project theo ngôn ngữ lập trình tương ứng.

Trong bài tập lớn này, sử dụng ngôn ngữ JavaScript ta cài đặt thông qua công cụ quản lý thư viện *yarn*. Tại thư mục project mở terminal và gõ dòng lệnh:

“yarn add selenium-webdriver”



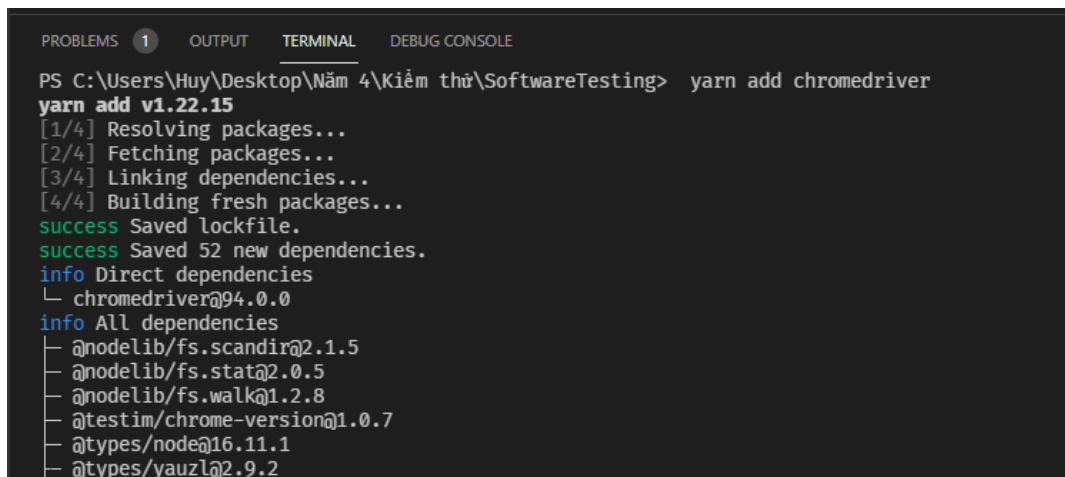
```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
C:\Users\Huy\Desktop\Năm 4\Kiểm thử\SoftwareTesting>yarn add selenium-webdriver
yarn add v1.22.15
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 25 new dependencies.
info Direct dependencies
└─ selenium-webdriver@4.0.0
info All dependencies
├─ balanced-match@1.0.2
├─ brace-expansion@1.1.11
├─ concat-map@0.0.1
├─ core-util-is@1.0.3
└─ fs.realpath@1.0.0
```

Hình 9. Hướng dẫn cài đặt Selenium WebDriver

Bước 2. Cài đặt trình điều khiển (driver) trình duyệt.

Với trình duyệt Chrome ta cài đặt thư viện *chromedriver*

“yarn add chromedriver”



```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\Huy\Desktop\Năm 4\Kiểm thử\SoftwareTesting> yarn add chromedriver
yarn add v1.22.15
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 52 new dependencies.
info Direct dependencies
└─ chromedriver@94.0.0
info All dependencies
├─ @node-lib/fs.scandir@2.1.5
├─ @node-lib/fs.stat@2.0.5
├─ @node-lib/fs.walk@1.2.8
├─ @testim/chrome-version@1.0.7
├─ @types/node@16.11.1
├─ @types/yauzl@2.9.2
```

Hình 10. Hướng dẫn cài đặt Selenium WebDriver

Bước 3. (Tùy chọn) Cài đặt và cấu hình Selenium Grid nếu muốn mở rộng quy mô kiểm thử.

IV. Giới thiệu dự án open source/ bài tập lớn

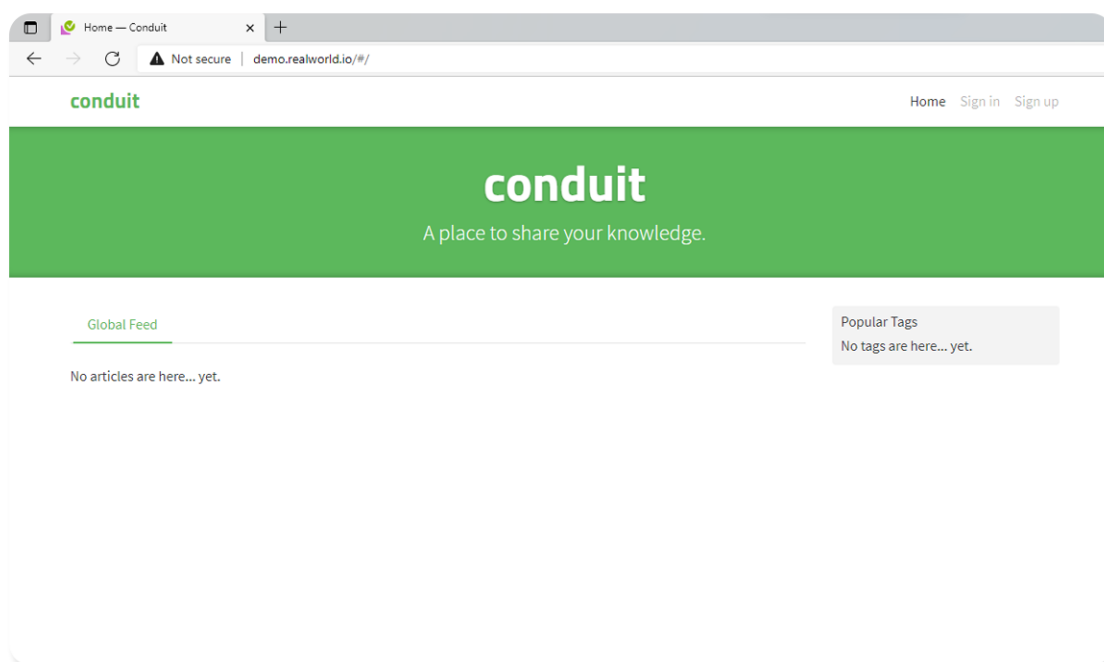
RealWorld: The mother of all demo apps

Repo: <https://github.com/gothinkster/realworld>

Demo: <https://demo.realworld.io/>

Video giới thiệu: <https://youtu.be/EL8DhHBqMAQ>

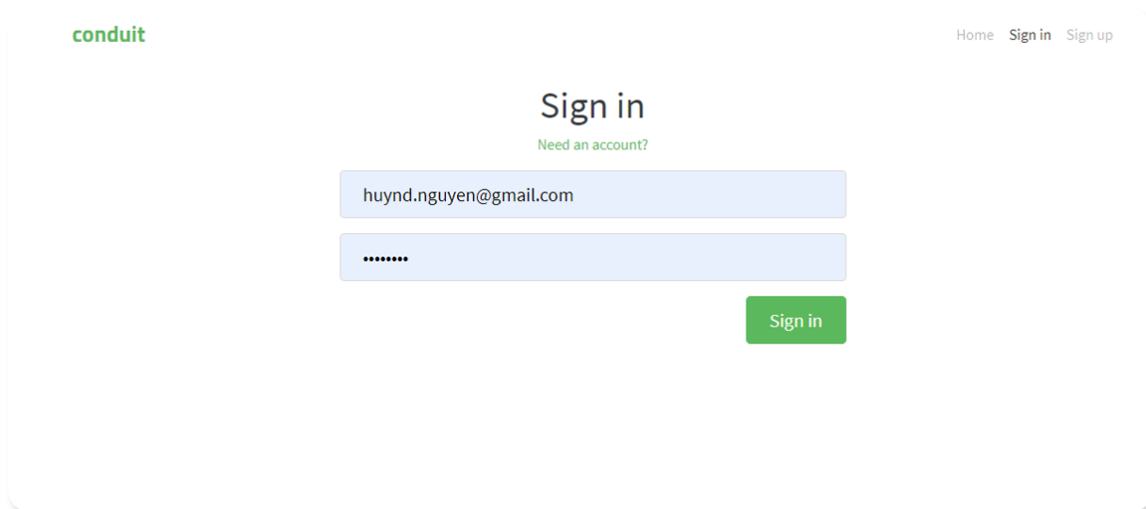
RealWorld là dự án xây dựng một trang blog xã hội Conduit (tương tự website medium.com) bằng nhiều ngôn ngữ / framework khác nhau.



Hình 11. Giao diện trang web Conduit

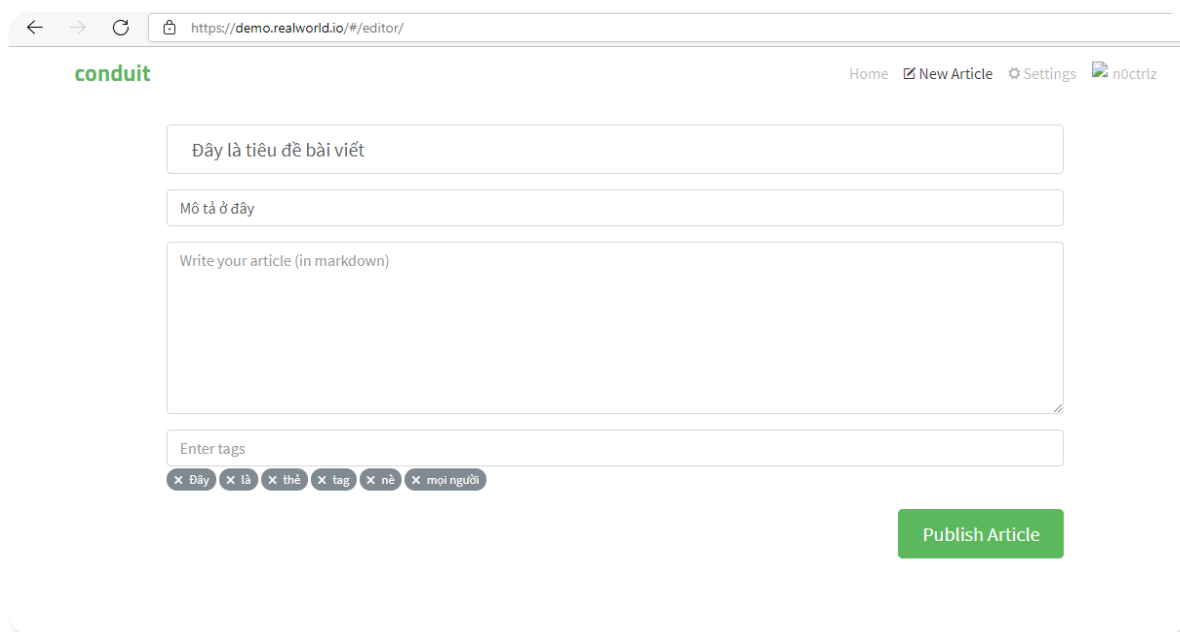
Các chức năng chung:

- Đăng nhập/Đăng xuất/ Đăng ký



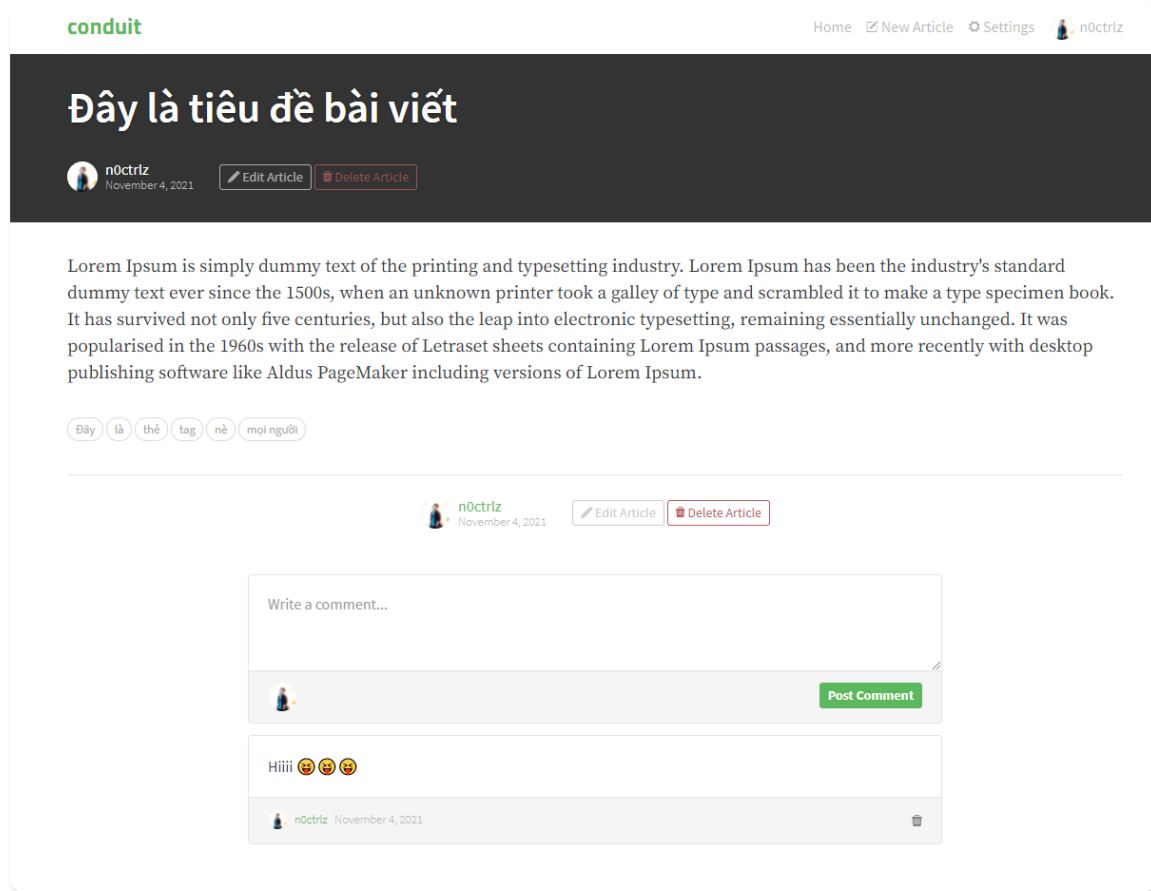
Hình 12. Giao diện trang Đăng nhập

- Đăng bài, chỉnh sửa hoặc xóa bài viết (của mình)



Hình 13. Giao diện trang Tạo bài viết

- Xem bài viết



Hình 14. Giao diện Xem bài viết

- Thêm bài viết yêu thích
- Bình luận/xóa bình luận
- Theo dõi người dùng khác
- Cài đặt trang

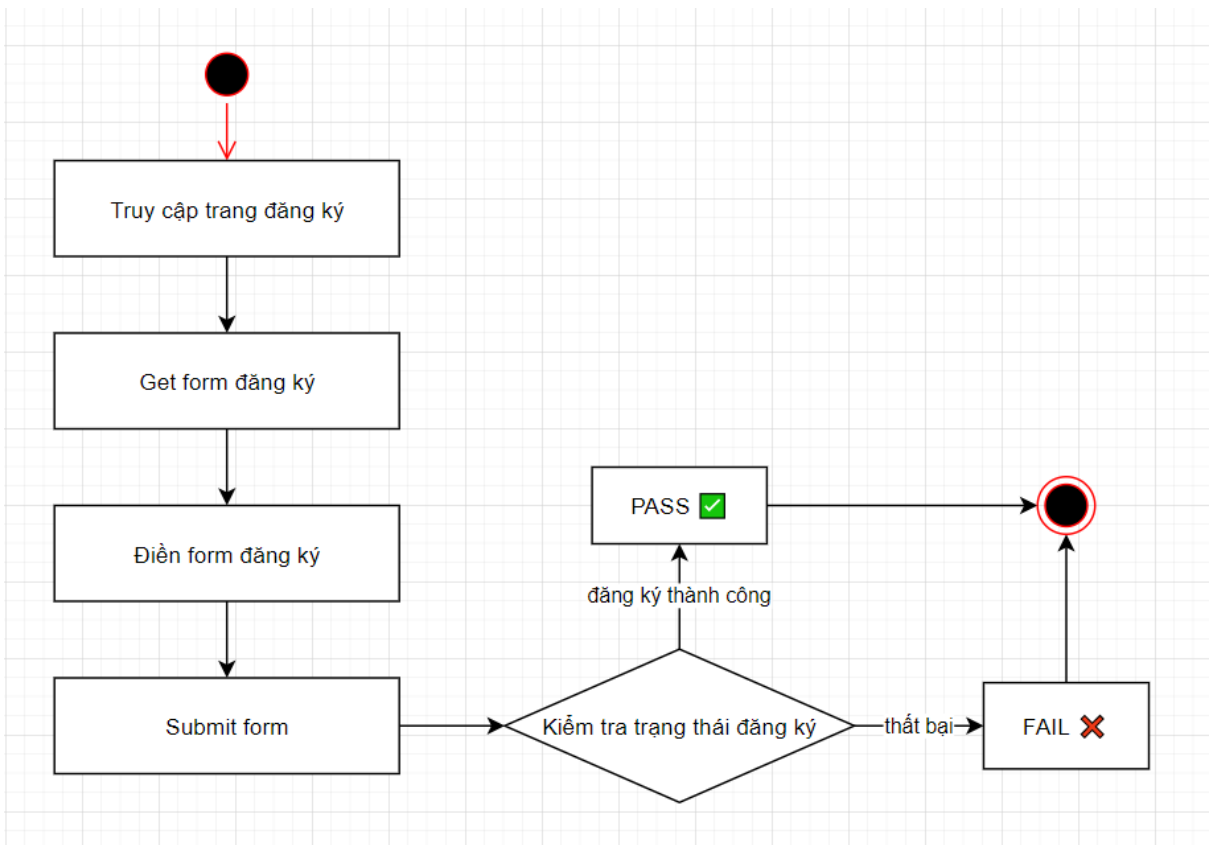
V. Báo cáo kết quả phân tích kiểm thử

1. Kiểm thử chức năng Đăng ký tài khoản

a. Mô tả bài toán

Kiểm thử chức năng Đăng ký tài khoản.

b. Sơ đồ luồng điều khiển



Hình 15. Sơ đồ luồng điều khiển kiểm thử chức năng Đăng ký tài khoản

c. Cài đặt kiểm thử

```

import { sleep } from './utils.js';
import { HOST_URL } from './const.js';
import { By, until } from 'selenium-webdriver';
import assert from 'assert';

const registerTest = async (driver, account) => {

  //--- Truy cập trang đăng ký ---

  await driver.get(HOST_URL + '/register');

  //--- Get ra form đăng ký ---

  const forms = await driver.findElements(By.className('form-control'));

  const [userNameInput, emailInput, passwordInput] = forms;
  const btn = await driver.findElement(By.className('btn'));

  //--- Điền form

  await userNameInput.click();
  await userNameInput.sendKeys(account.username);

  await emailInput.click();
  await emailInput.sendKeys(account.email);

  await passwordInput.click();
  await passwordInput.sendKeys(account.password);

  //--- Submit form

  await sleep(1000);
  btn.click();

  //--- Kiểm tra đã đăng ký thành công hay chưa

  await driver.wait(until.elementLocated(By.className('ion-gear-a')), 10000);

  const isSuccess = await driver.findElement(By.className('ion-gear-a'));


  assert.equal(isSuccess !== null, true);

};

export default registerTest;

```

Hình 16. Hàm kiểm thử chức năng Đăng ký tài khoản

d. Kết quả kiểm thử: Pass 

```

Testing Realworld

DevTools listening on ws://127.0.0.1:51054/devtools/browser/b7475ead-8eb0-4a80-9eb1-5577adde55c0
✓ create new account (2337ms)

1 passing (3s)

Done in 4.22s.

```

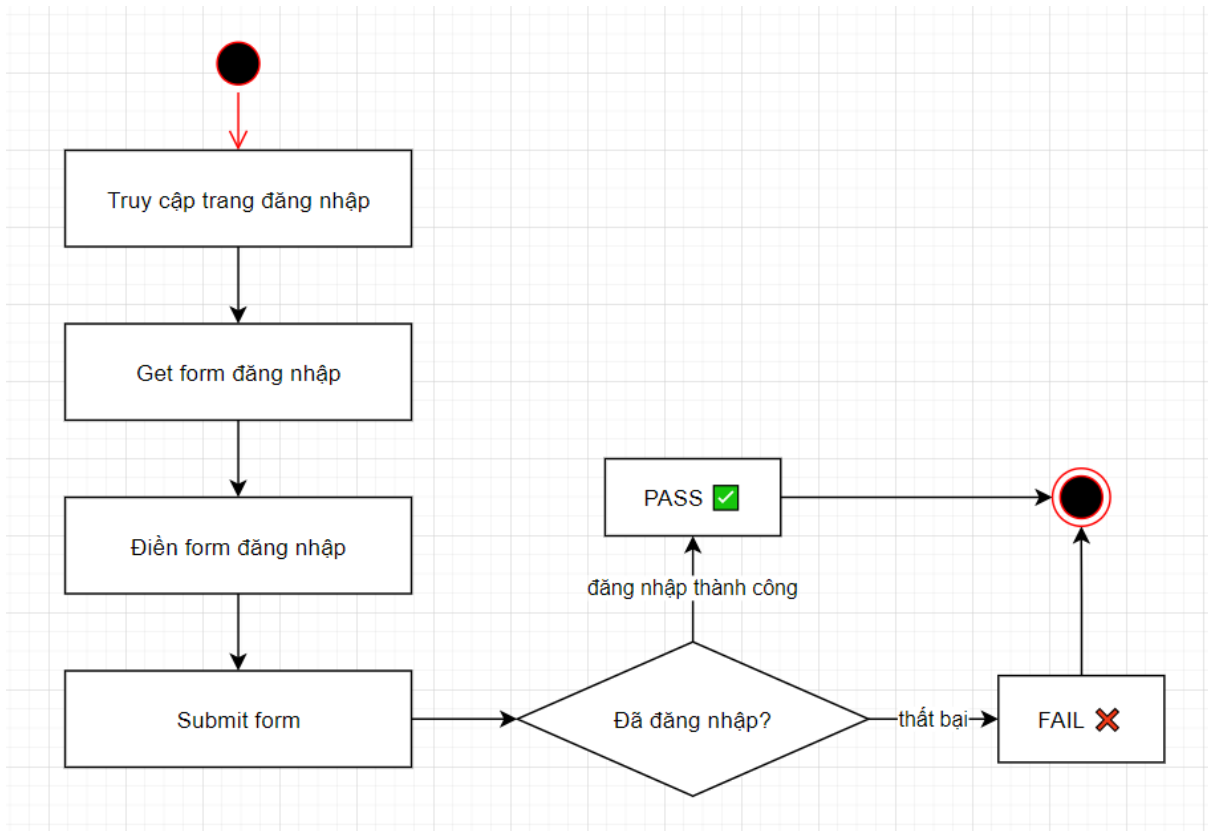
Hình 17. Kết quả kiểm thử chức năng Đăng ký tài khoản

2. Kiểm thử chức năng Đăng nhập

a. Mô tả bài toán

Kiểm thử chức năng Đăng nhập.

b. Sơ đồ luồng điều khiển



Hình 18. Sơ đồ luồng điều khiển kiểm thử chức năng Đăng nhập

c. Cài đặt kiểm thử

```

import { HOST_URL } from './const.js';
import { sleep } from './utils.js';
import { Key, By, until } from 'selenium-webdriver';
import assert from 'assert';

const loginTest = async (driver, account) => {

  // --- Truy cập trang đăng nhập

  await driver.get(`${HOST_URL}/login`);

  // --- Get ra form đăng nhập

  const inputs = await driver.findElements(By.className('form-control'));

  // --- Điền form đăng nhập

  await inputs[0].click();
  await inputs[0].sendKeys(account.email);
  await inputs[1].click();
  await inputs[1].sendKeys(account.password);

  // --- Đăng nhập

  await inputs[1].sendKeys(Key.RETURN);

  // --- Kiểm tra đã đăng nhập thành công hay chưa

  await driver.wait(until.elementLocated(By.className('ion-gear-a')), 10000);

  const isSuccess = await driver.findElement(By.className('ion-gear-a'));
  assert.equal(isSuccess !== null, true);
};

export default loginTest;

```

Hình 19. Hàm kiểm thử chức năng Đăng nhập

d. Kết quả kiểm thử: Pass 

```

Testing Realworld

DevTools listening on ws://127.0.0.1:49258/devtools/browser/7a535e0f-6598-4754-85c1-bbf5c4ae7b08
  ✓ create new account (2302ms)
  ✓ login realworld (353ms)

2 passing (3s)

```

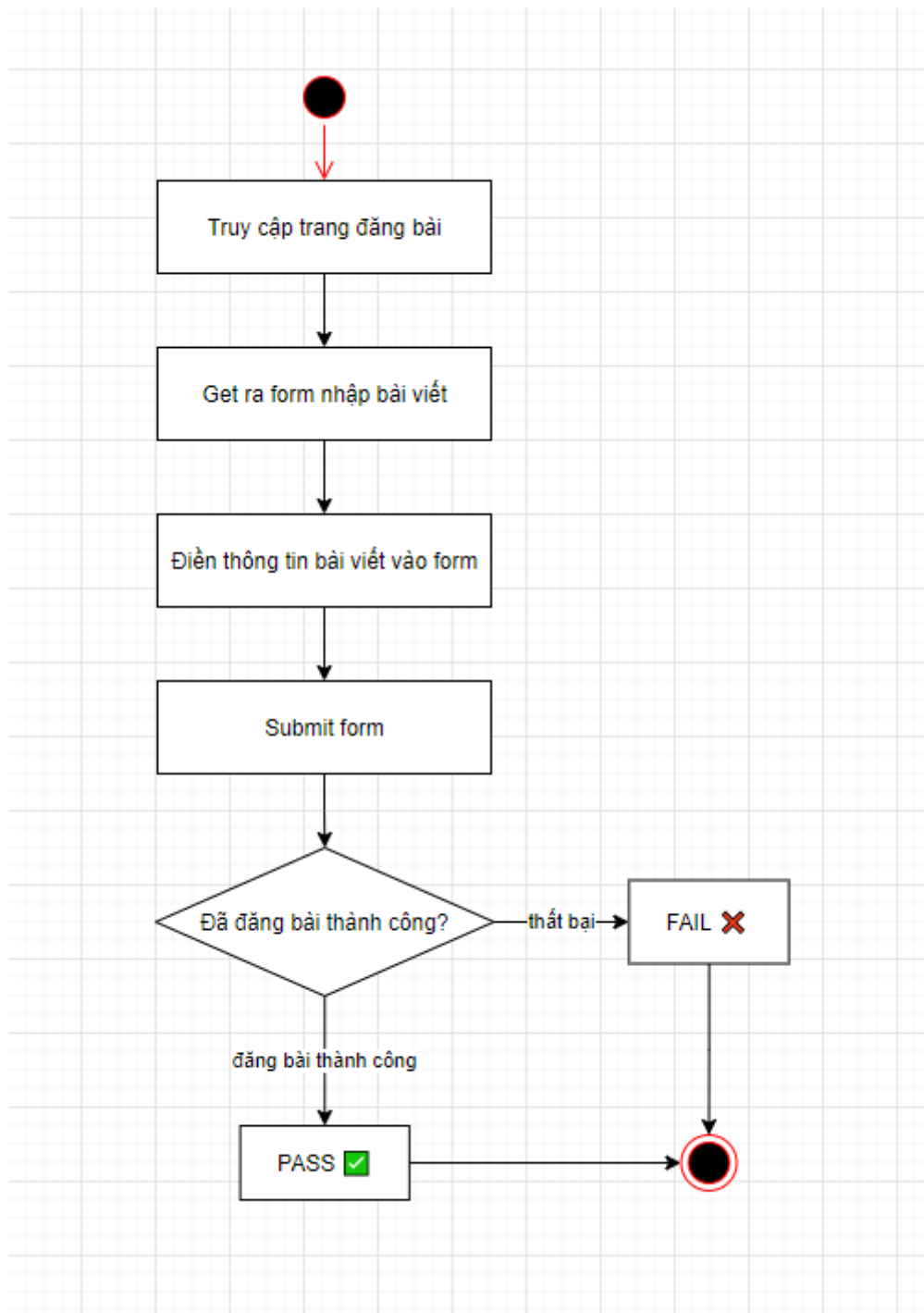
Hình 20. Kết quả kiểm thử chức năng Đăng nhập

3. Kiểm thử chức năng Tạo bài viết

a. Mô tả bài toán

Kiểm thử chức năng Tạo bài viết.

b. Sơ đồ luồng điều khiển



Hình 21. Sơ đồ luồng điều khiển kiểm thử chức năng Tạo bài viết

c. Cài đặt kiểm thử

```
import { sleep } from './ultis.js';
import { HOST_URL } from './const.js';
import { Key, By, until } from 'selenium-webdriver';
import faker from 'faker';
import assert from 'assert';

const createPost = async (driver) => {
  faker.locale = 'en_US';

  // --- Truy cập trang đăng bài

  await driver.get(`${HOST_URL}/editor`);

  // --- Get ra form nhập bài viết

  const forms = await driver.findElements(By.className('form-control'));
  const btn = await driver.findElement(By.className('btn'));
  const [titleInput, descriptionInput, bodyInput, tagInput] = forms;

  // --- Điền thông tin bài viết vào form

  await titleInput.click();
  await titleInput.sendKeys(faker.lorem.sentence());

  await descriptionInput.click();
  await descriptionInput.sendKeys(faker.lorem.sentences(2));

  await bodyInput.click();
  await bodyInput.sendKeys(faker.lorem.paragraph());

  await tagInput.click();
  await tagInput.sendKeys(
    faker.lorem.word(),
    Key.RETURN,
    faker.lorem.word(),
    Key.RETURN,
    faker.lorem.word(),
    Key.RETURN,
    faker.lorem.word(),
    Key.RETURN
  );


  btn.click();
  await driver.wait(until.elementLocated(By.className('article-page')), 10000);

  // --- Kiểm tra thành công

  const isSuccess = await driver.findElement(By.className('article-page'));
  assert.equal(isSuccess !== null, true);
};

export default createPost;
```

Hình 22. Hàm kiểm thử chức năng Tạo bài viết

d. Kết quả kiểm thử: Pass 

```
Testing Realworld
DevTools listening on ws://127.0.0.1:62353/devtools/browser/55d3e7b9-c62c-41d6-a9de-75d65434c673
✓ create new account (2745ms)
✓ login realworld (337ms)
✓ create post (1150ms)

3 passing (5s)

Done in 6.17s.
```

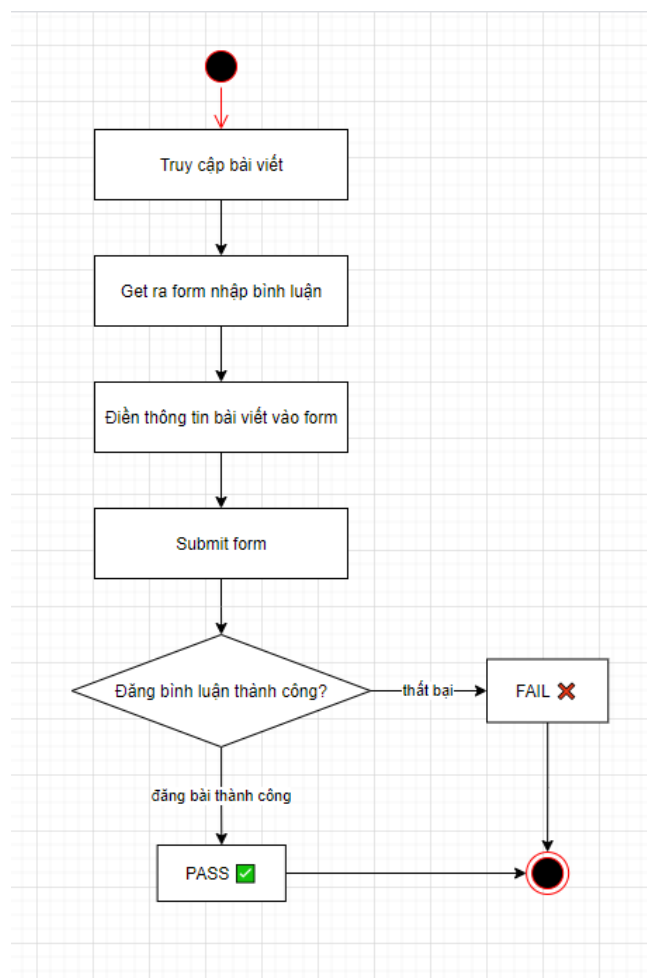
Hình 23. Kết quả kiểm thử chức năng Tạo bài viết

4. Kiểm thử chức năng Bình luận bài viết

a. Mô tả bài toán

Kiểm thử chức năng bình luận bài viết

b. Sơ đồ luồng điều khiển



Hình 24. Sơ đồ luồng điều khiển kiểm thử chức năng Bình luận

c. Cài đặt kiểm thử

```
import { sleep } from './utils.js';
import { By, until } from 'selenium-webdriver';
import faker from 'faker';
import assert from 'assert';

const commentPost = async (driver) => {

  // --- Get form điền bình luận và nút submit

  const textArea = await driver.findElement(By.xpath('//textarea'));
  const btn = await driver.findElement(By.xpath('//button[@type='submit']`));

  // --- Điền bình luận

  await textArea.click();
  await textArea.sendKeys(faker.lorem.sentence());

  // --- Submit

  await btn.click();
  await driver.wait(until.elementLocated(By.className('ion-trash-a')), 10000);

  // --- Kiểm tra thành công

  const isSuccess = await driver.findElement(By.className('ion-trash-a'));
  assert.equal(isSuccess !== null, true);

};

export default commentPost;
```

Hình 25. Hàm kiểm thử chức năng Bình luận

d. Kết quả kiểm thử

```
PS C:\Users\Huy\Desktop\Năm 4\Kiểm thử\SoftwareTesting> yarn run test
yarn run v1.22.15
$ mocha --recursive --timeout 10000 main/index.js

Testing Realworld

DevTools listening on ws://127.0.0.1:58668/devtools/browser/c0f0eb91-aec1-49b8-b92b-5ecf6daeb686
  ✓ create new account (2308ms)
  ✓ login realworld (326ms)
  ✓ create post (1808ms)
  ✓ comment post (162ms)

4 passing (5s)
```

Hình 26. Kết quả kiểm thử chức năng Bình luận

VI. So sánh - Tổng kết

Tóm lại, Selenium là một bộ kiểm thử tự động mã nguồn mở miễn phí cho các ứng dụng web trên nhiều trình duyệt và hệ điều hành khác nhau. Selenium khá giống với HP Quick Test Pro (QTP) nhưng Selenium chỉ tập trung vào việc tự động hóa các ứng dụng dựa trên web.

So sánh Selenium và QTP

Nội dung so sánh	Selenium	QTP
Chi phí bản quyền	Miễn phí (mã nguồn mở)	Mất phí (Chi phí rất lớn)
Hệ điều hành hỗ trợ	Windows, Mac OS, Linux,...	Windows
Loại ứng dụng hỗ trợ	Ứng dụng web	Ứng dụng web và desktop
Trình duyệt hỗ trợ	FireFox, Google Chrome, IE, Safari, Opera,...	FireFox, Google Chrome, IE
Ngôn ngữ hỗ trợ	Java, Ruby, C#, Python, Perl,...	VBScript, JavaScript
Plug-ins	Miễn phí	Mất phí
Yêu cầu kỹ năng để sử dụng	Người dùng cần có kỹ năng lập trình tốt	Yêu cầu thấp hơn
Khả năng tạo kịch bản kiểm thử	Tốt	Hoàn hảo
Môi trường phát triển kiểm thử	Có thể phát triển trên nhiều trình soạn thảo văn bản	Chỉ có thể phát triển trong QTP
Hỗ trợ kỹ thuật	Không có hỗ trợ kỹ thuật chính thức	Hỗ trợ kỹ thuật một cách chuyên nghiệp

Tiêu thụ tài nguyên khi kiểm thử	Tiêu thụ tài nguyên thấp	Tiêu thụ tài nguyên cao
Kho lưu trữ đối tượng, Giải pháp phục hồi	Không hỗ trợ	Có kho lưu trữ tích hợp sẵn và kế hoạch phục hồi
Kiểm soát trình duyệt	Không	Có
Tạo báo cáo kiểm thử	Dựa vào các công cụ bên ngoài để tạo báo cáo	Tích hợp sẵn trong công cụ

Selenium sẽ là lựa chọn hoàn hảo nếu:

- Muốn sử dụng công cụ miễn phí.
- Yêu cầu kiểm thử trên nhiều trình duyệt và hệ điều hành khác nhau.
- Hỗ trợ kiểm thử bằng nhiều loại ngôn ngữ.

Bên cạnh những ưu điểm trên thì Selenium cũng có những hạn chế nhất định:

- Chỉ có thể kiểm thử trên trình duyệt.
- Tốn thời gian cho việc cài đặt và tích hợp cho dự án mới.
- Người dùng cần có kỹ năng lập trình tốt để cài đặt và tích hợp với các framework khác.
- Hỗ trợ chậm từ cộng đồng do không có bộ phận hỗ trợ chính thức.

VII. Nguồn tham khảo

1. <https://www.selenium.dev/>
2. <https://tech.cybozu.vn/>
3. <https://viettuts.vn/selenium>
4. <https://itnavi.com.vn/blog>
5. <https://viblo.asia/>
6. <https://vananhthooo.wordpress.com/>
7. <https://topdev.vn/>