**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

BÁO CÁO BÀI TẬP LỚN HỌC PHẦN:

KIỂM THỬ PHẦN MỀM

**NGHIÊN CỨU VÀ ỨNG DỤNG CÔNG CỤ SELENIUM IDE VÀO ỨNG DỤNG KIỂM THỬ WEBSITE** [**TOPCV.VN**](http://taingheviet.com/)

|  |  |
| --- | --- |
| Giáo viên: | Ths.Nguyễn Thái Cường |
| Nhóm - Lớp: | 8 |
| Thành viên: | Đỗ Văn Hưng |
|  | Lê Tuấn Hưng  Chu Văn Quý  Trần Tiến Thịnh  Nguyễn Dũng Tuấn  Lớp: 20232IT6084004 - K16 |
| Hà Nội,T6/2024 |  |
|  |  |

**Mục lục**

[LỜI MỞ ĐẦU 4](#_Toc168299419)

[Chương 1. Giới thiệu kiểm thử phần mềm 5](#_Toc168299420)

[1.1 Tổng quan về kiểm thử phần mềm 5](#_Toc168299421)

[1.2 Quy trình kiểm thử phần mềm 6](#_Toc168299422)

[1.3 Các cấp độ kiểm thử 8](#_Toc168299423)

[1.4 Các loại hình kiểm thử 12](#_Toc168299424)

[1.5 Các kỹ thuật kiểm thử 13](#_Toc168299425)

[Chương 2. Giới thiệu phần mềm 17](#_Toc168299426)

[2.1 Yêu cầu đề tài 17](#_Toc168299427)

[2.2 Phân tích thiết kế hệ thống 19](#_Toc168299428)

[2.2.1. Phân tích use case Đăng ký 19](#_Toc168299429)

[2.2.2. Phân tích use case Đăng nhập 21](#_Toc168299430)

[2.2.3. Phân tích use case Tìm kiếm công việc 22](#_Toc168299431)

[2.2.4. Phân tích use case Cập nhật thông tin cá nhân 24](#_Toc168299432)

[2.2.5. Phân tích use case Ứng tuyển 25](#_Toc168299433)

[Chương 3. Kiểm thử phần mềm 28](#_Toc168299434)

[3.1 Lập kế hoạch 28](#_Toc168299435)

[3.1.1 Giới thiệu chung về dự án 28](#_Toc168299436)

[3.1.2 Phạm vi kiểm thử 28](#_Toc168299437)

[3.1.3 Các tiêu chí chấp nhận sản phẩm 29](#_Toc168299438)

[3.1.4 Các tiêu chí chấp nhận sản phẩm 29](#_Toc168299439)

[3.1.5 Lịch trình công việc 30](#_Toc168299440)

[3.1.6 Sản phẩm bàn giao 30](#_Toc168299441)

[3.2 Thực hiện kiểm thử 30](#_Toc168299442)

[3.2.1 Lê Tuấn Hưng-Đăng nhập 30](#_Toc168299443)

[3.2.2 Trần Tiến Thịnh - Tìm kiếm công việc 34](#_Toc168299444)

[3.2.3 Chu Văn Quý - Đăng ký 38](#_Toc168299445)

[3.2.4 Ứng tuyển - Nguyễn Dũng Tuấn 44](#_Toc168299446)

[3.2.5 Cập nhật thông tin cá nhân - Đỗ Văn Hưng 49](#_Toc168299447)

[TỔNG KẾT 54](#_Toc168299448)

[Tài liệu tham khảo 55](#_Toc168299449)

LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin không ngừng phát triển, việc đảm bảo chất lượng phần mềm, đặc biệt là các ứng dụng web, trở thành một yếu tố then chốt đối với sự thành công của các sản phẩm công nghệ. Kiểm thử phần mềm không chỉ giúp phát hiện sớm các lỗi tiềm ẩn mà còn nâng cao trải nghiệm người dùng, đảm bảo sự ổn định và an toàn của hệ thống.

Trong phạm vi nghiên cứu này, chúng tôi tập trung vào việc ứng dụng công cụ Selenium IDE, một trong những công cụ kiểm thử tự động phổ biến và mạnh mẽ hiện nay, vào quá trình kiểm thử trang web TopCV.VN. TopCV.VN là một nền tảng cung cấp dịch vụ tạo và chia sẻ hồ sơ xin việc trực tuyến, với lượng người dùng ngày càng tăng và yêu cầu về hiệu năng, tính năng cũng như trải nghiệm người dùng ngày càng cao.

Bằng cách áp dụng công cụ này vào quá trình kiểm thử TopCV.com, nghiên cứu không chỉ mong muốn nâng cao hiệu quả kiểm thử mà còn góp phần đảm bảo chất lượng dịch vụ của trang web.

Mục tiêu của nghiên cứu này là cung cấp cái nhìn tổng quan về cách thức triển khai và lợi ích của việc sử dụng Selenium IDE trong kiểm thử tự động. Đồng thời, chúng tôi sẽ trình bày chi tiết các bước thực hiện kiểm thử, các kết quả đạt được cũng như những kinh nghiệm rút ra từ quá trình nghiên cứu. Hy vọng rằng, nghiên cứu này sẽ là nguồn tham khảo hữu ích cho các nhà phát triển và kiểm thử phần mềm, giúp họ tối ưu hóa quy trình kiểm thử và nâng cao chất lượng sản phẩm của mình.

1. Giới thiệu kiểm thử phần mềm

1.1 Tổng quan về kiểm thử phần mềm

* Kiểm thử phần mềm (software testing) là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, đúng và đầy đủ theo yêu cầu của khách hàng, yêu cầu của sản phẩm đã đặt ra. Nó cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm điều này cho phép đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.
* Kiểm thử phần mềm đóng vai trò rất quan trọng:

+ Kiểm thử phần mềm là hoạt động đảm bảo chất lượng phần mềm và mang tính sống còn trong các dự án sản xuất phần mềm. Vì vậy nó đã trở thành quy trình bắt buộc trong các dự án phần mềm hiện nay.

+ Kiểm thử phần mềm để tránh những rủi ro, lỗi phát sinh trong suốt quá trình tạo ra sản phẩm.

+ Lỗi càng phát hiện ra sớm càng giúp tránh được rủi ro và chi phí.

* Mục đích của kiểm thử phần mềm:

+ Kiểm thử phần mềm để đánh giá phần mềm có đạt yêu cầu mong đợi hay có sai sót nào không?

+ Phần mềm có làm việc như mong muốn không?

+ Phần mềm có giải quyết được yêu cầu của khách hàng không?Nó làm được gì mà người dùng mong đợi?

+ Người dùng có thích nó không?

+ Nó có tương thích với các hệ thống khác của chúng ta hay không?

* Nhiều công ty phát triển phần mềm thường bỏ qua bước này vì ngân sách eo hẹp và cho rằng nó sẽ không dẫn đến hậu quả lớn. Nhưng để tạo những trải nghiệm tốt nhất cho khách hàng, chất lượng sản phẩm cần phải được đặt lên hàng đầu. Và vì vậy, việc kiểm thử sản phẩm để tìm lỗi là điều gần như bắt buộc. Doanh nghiệp chỉ có thể mang đến giá trị cho khách hàng khi sản phẩm cung cấp được coi là lý tưởng. Và để đạt được điều đó, các công ty phải đảm bảo rằng người dùng không gặp phải bất kỳ vấn đề nào khi sử dụng sản phẩm của mình. Cách tốt nhất để làm điều đó là tạo ra sản phẩm không có lỗi.
  1. Quy trình kiểm thử phần mềm
* Quy trình kiểm thử phần mềm xác định các giai đoạn, pha trong kiểm thử phần mềm.

A blue rectangular arrows with white text

Description automatically generated

*Hình 1. Quy trình kiểm thử phần mềm*

|  |  |  |  |
| --- | --- | --- | --- |
| **Quy trình** | **Đầu vào** | **Các hoạt động** | **Đầu ra** |
| Phân tích yêu cầu | Tài liệu SRS, tài liệu thiết kế, bản prototype | Đọc hiểu, nghiên cứu phân tích các yêu cầu có trong các bản tài liệu Đưa ra các câu hỏi còn thắc mắc về yêu cầu phần mềm với BA, team, leader, khách hàng để hiểu rõ hơn về yêu cầu sản phẩm | File Q & A |
| Lập kế hoạch | Các tài liệu đã được cập nhật thông qua file Q & A trong giai đoạn phân tích yêu cầu | Xác định phạm vi kiểm thử: thời gian, lịch trình cho các công việc. Xác định phương pháp tiếp cận. Xác định nguồn lực: con người và thiết bị. Lên kế hoạch thiết kế công việc test: các chức năng cần kiểm thử, cái nào cần thực hiện trước, sau, ai là người thực hiện... | Test plan, checklist |
| Thiết kế kiểm thử | Test plan, smoke test case, test data | Thực thi các smoke test case để kiểm tra môi trường kiểm thử đã sẵn sàng cho việc test chưa | Môi trường đã được chuẩn bị sẵn sàng cho việc test và các kết quả của smoke test case |
| Thực hiện kiểm thử | Test design, test case, check list, test data, test automation script | Thực hiện test theo kịch bản kiểm thử. So sánh kết quả thực tế với mong đợi và log bug lên tool quản lý lỗi, theo dõi quá trình xử lý lỗi. | Test results, defect reports |
| Kết thúc | Tất cả các tài liệu được tổng hợp từ giai đoạn đầu tiên | Tổng kết báo cáo kết quả về việc thực thi test, chức năng nào hoàn thành/ chức năng chưa hoàn thành, lỗi còn nhiều ở chức năng nào, dev nào còn nhiều lỗi, lỗi có nghiêm trọng hay không... | Test report, test results final |

* 1. Các cấp độ kiểm thử
* Trước khi phát hành một phần mềm nào đó, chúng đã phải trải qua một quá trình kiểm tra kỹ lưỡng để đảm bảo rằng phần mềm sẽ hoạt động mượt mà, ổn định theo đúng chức năng được thiết kế.
* Có 04 mức độ kiểm thử phần mềm chính cần hoàn thành trước khi đưa phần mềm vào sử dụng: kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống, kiểm thử chấp nhận.
* Đầu tiên, chúng ta sẽ đến với kiểm thử đơn vị (Unit testing)

+ Unit testing, còn được gọi là Component testing, mức độ kiểm thử phần mềm này thường do Developer phụ trách, họ sẽ đi kiểm tra các module, các hàm, các phương thức, các lớp,… mà họ viết ra nhằm gia tăng sự tin cậy cho các chức năng mà mình viết.

+ Kiểm thử đơn vị nằm trong phạm vi của Kiểm thử hộp trắng (White-box testing), tức là kiểm tra code bên trong của một chức năng hoặc hệ thống để xem chức năng hoặc hệ thống đó được viết đúng chuẩn code hay chưa, đoạn code đó khi chạy hiệu năng có tốt hay không, có nhanh hay không, có tốn tài nguyên hay không,..

+ Tuy nhiên, kiểm thử theo phương pháp này không thể tìm được mọi lỗi của ứng dụng. Không thể đánh giá mọi tình huống có thể xảy ra trong chương trình. Nó bị giới hạn bởi các tình huống và bộ dữ liệu test mà developer sử dụng để kiểm thử chương trình.

* **Kiểm thử tích hợp (Integration testing)**

+ Kiểm thử tích hợp là kiểm thử sự tương tác giữa các chức năng với nhau trong hệ thống và được thực hiện bởi Tester. Ví dụ: sau khi đã unit test chức năng đăng nhập và chức năng đăng ký thì ta có thể tiến hành kiểm thử tích hợp của 2 chức năng này để xem chúng có tương tác tốt với nhau không, sau khi đăng ký thành công thì ta có thể tiến hành đăng nhập bằng tài khoản đã đăng ký xem có thực hiện được không.

+ Một ví dụ khác: Sau khi unit test cho các chức năng con trong chức năng đăng ký môn học như đăng nhập, đăng ký tài khoản, kiểm tra điều kiện tiên quyết, kiểm tra học phí,.. thì ta kiểm tra sự tích hợp giữa các chức năng này bằng cách tiến hành đăng ký một môn học để xem sự tương tác giữa các chức năng này có thực hiện được không, có trơn tru không, có bị mất liên kết chỗ nào không,…

+ Kiểm thử tích hợp có hai cách tiếp cận phổ biến: Tích hợp từ dưới đi lên (Bottom-up integration) và tích hợp đi trên đi xuống (Top-down integration).

+ Tích hợp từ dưới đi lên: Việc kiểm thử được bắt đầu bằng unit test, sau đó sẽ đi đến mức cao hơn, đó là kết hợp các thành phần và chức năng của hệ thống.

A diagram of a system

Description automatically generated

*Hình 2. Tích hợp từ dưới đi lên*

+ **Tích hợp từ trên đi xuống**: Những module cấp cao nhất được thử nghiệm đầu tiên rồi đến các module thấp hơn được thử nghiệm.

*A diagram of a diagram

Description automatically generated*

*Hình 3. Tích hợp từ trên đi xuống*

Trong một môi trường phát triển phần mềm hiện đại, kiểm thử từ dưới lên thường được thực hiện đầu tiên và sau đó là kiểm thử từ trên xuống. Quá trình này được kết thúc sau nhiều vòng kiểm thử trên ứng dụng hoàn chỉnh, tốt nhất là sử dụng các kịch bản kiểm thử dựa trên các tình huống thực tế.

* **Kiểm thử hệ thống (System testing)**

*+* Kiểm thử hệ thống là kiểm thử một hệ thống đã hoàn thành, đã tích hợp đầy đủ các chức năng nhằm kiểm tra xem hệ thống phần mềm đó có đáp ứng đầy đủ các yêu cầu chức năng theo bản đặc tả yêu cầu phần mềm (SRS) hay không. Người thực hiện test level này thường là Tester.

*+* Kiểm thử hệ thống thuộc phạm vi Kiểm thử hộp đen (tức là Tester chỉ quan tâm đầu vào và kết quả mong đợi ở đầu ra mà không cần kiểm tra code bên trong được viết như thế nào).

*+* Dưới đây là một số loại kiểm thử thường được thực hiện trong System Test:

* Kiểm thử chức năng (**Functional Test**): Là kiểm thử toàn bộ hệ thống, đảm bảo hệ thống hoạt động đúng theo yêu cầu được đưa ra trước đó
* Kiểm thử hiệu năng (**Performance Test**): Là kiểm tra sự tuân thủ của hệ thống với các yêu cầu được chỉ định về hiệu năng. Xác định những thuộc tính chất lượng của hệ thống như khả năng mở rộng, độ tin cậy…
* Kiểm thử cơ sở dữ liệu ( **Database Test**): Là kiểm tra dữ liệu hiển thị trên hệ thống có giống với dữ liệu trong cơ sở dữ liệu hay không?
* Kiểm thử khả năng bảo mật ( **Security Test**): Là kiểm tra hệ thống được bảo vệ an toàn, không bị đánh cắp dữ liệu, thông tin trước các tấn công từ bên ngoài.
* Kiểm thử tính khả dụng (**Usability Test**): Kiểm tra tính thân thiện với người dùng và tính dễ sử dụng của hệ thống.
* Kiểm tra tính tương thích ( **Compatibility Test**) : Là kiểm tra xem hệ thống có tương thích với các yếu tố khác của hệ thống mà nó sẽ hoạt động hay không? (Ví dụ: Trình duyệt, hệ điều hành, phần cứng).
* Kiểm tra khả năng phục hồi ( **Recovery Test**): Là kiểm tra hệ thống có khả năng khôi phục trạng thái ổn định khi gặp các sự cố bất thường không.
* **Kiểm thử chấp nhận (Acceptance testing)**

**+** Mức độ kiểm thử phần mềm cuối cùng chính là Acceptance Test (Kiểm thử chấp nhận) – kiểm tra xem hệ thống có đáp ứng đúng nhu cầu và mong đợi của khách hàng hay không.

+ Kiểm thử chấp nhận thường là trách nhiệm của người dùng hoặc khách hàng. Trong kiểm thử hệ thống, khách hàng sẽ kiểm tra xem phần mềm được viết có hoạt động đúng như mong đợi của mình không, có đảm bảo tính tiện dụng, hiệu suất hoạt động có như mong đợi không, có bảo mật tốt hay không,….

+ Tìm lỗi không phải là trọng tâm chính trong kiểm thử chấp nhận, vì việc tìm lỗi đã được đội Developer và Tester thực hiện trong các giai đoạn kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống rồi.

+ Acceptance test gồm 2 loại kiểm thử là:

* Alpha Test, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa.
* Beta Test, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa. Lưu ý không nhất thiết phải thực hiện tất cả các loại kiểm tra nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tuỳ khả năng và thời gian cho phép của dự án, khi lập kế hoạch, trưởng dự án sẽ quyết định áp dụng những loại kiểm tra nào.
  1. Các loại hình kiểm thử
* Kiểm thử phần mềm là một khâu không thể thiếu khi phát triển bất kỳ hệ thống, website hay ứng dụng nào. Kiểm thử giúp đảm bảo hệ thống không xảy ra lỗi và vận hành đúng chức năng mong đợi ban đầu. Trong phần này, nhóm sẽ chỉ ra các loại kiểm thử phần mềm thông dụng nhất trong công nghệ thông tin.
* Ở cấp độ cao, kiểm thử thường được chia làm 2 loại: Kiểm thử thủ công và kiểm thử tự động, hay còn được đọc ngắn gọn là “manual test” và “auto test”.
* Kiểm thử thủ công:

+ Kiểm thử thủ công thường được thực hiện bằng chính bản thân kiểm thử viên (tester). Họ sẽ tương tác với ứng dụng hoặc phần mềm và API bằng công cụ thích hợp. Từ đó tester tìm ra điểm không phù hợp hay các lỗi của hệ thống. Cách kiểm thử truyền thống này thường  tốn kém vì nó yêu cầu môi trường kiểm thử. Cùng với đó, việc tự thực hiện các thao tác kiểm thử có thể dễ xảy ra lỗi do con người. Vì người kiểm thử có thể mắc lỗi chính tả hoặc bỏ qua các bước trong tập lệnh kiểm thử (test script).

* Kiểm thử tự động:

+ Ngược lại, kiểm thử tự động được thực hiện bởi máy móc, thực thi tập lệnh kiểm thử đã được viết trước. Các lệnh kiểm thử này có thể khác nhau rất nhiều về độ phức tạp. Từ việc kiểm tra các đơn vị nhỏ nhất trong mã nguồn như method, class đến việc đảm bảo rằng việc thực hiện một chuỗi các hành động phức tạp trong giao diện người dùng sẽ dẫn đến kết quả giống nhau. Nhiều người cho rằng phương pháp này mạnh mẽ và đáng tin cậy hơn so với kiểm thử thủ công. Nhưng chất lượng của các lệnh kiểm thử tự động phụ thuộc việc các tập lệnh kiểm thử được viết ra có tốt hay không.

* 1. Các kỹ thuật kiểm thử
* Kỹ thuật kiểm thử phần mềm giúp bạn thiết kế các trường hợp kiểm thử tốt hơn. Vì kiểm thử toàn diện là không thể nên kỹ thuật kiểm tra thủ công sẽ giúp giảm số lượng các trường hợp kiểm thử được thực thi trong khi tăng phạm vi kiểm thử. Chúng giúp xác định các điều kiện kiểm tra khó nhận biết.
* Dưới đây là các loại kỹ thuật kiểm thử thông dụng:
* **Phân vùng tương đương:**

+ Phân vùng lớp tương đương cho phép bạn phân chia tập hợp các điều kiện kiểm tra thành một phân vùng nên được coi là giống nhau.

+ Phương pháp kiểm thử phần mềm này chia miền đầu vào của chương trình thành các lớp dữ liệu mà từ đó các trường hợp kiểm thử nên được thiết kế.

+ Với các giá trị đầu vào chia thành các vùng tương đương:

* + Vùng tương đương hợp lệ: tập hợp các giá trị kiểm thử thỏa mãn điều kiện của hệ thống
  + Vùng tương đương không hợp lệ: Tập hợp các giá trị kiểm thử mô trạng thái khác của hệ thống: sai, thiếu, không đúng,...

+ Mục đích : Giảm đáng kể số lượng test case cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện.

+ Thiết kế Test-case bằng phân lớp tương đương tiến hành theo 2 bước:

* + Xác định các lớp tương đương
  + Xác định các ca kiểm thử

+ Nguyên tắc:

* 1 lớp các giá trị lớn hơn
* 1 lớp các giá trị nhỏ hơn
* n lớp các giá trị hợp lệ

+ Ví dụ: Thiết kế testcase cho ô text chỉ cho nhập số nguyên với độ dài ký tự thuộc [1-10] hoặc [20-30]

+ Với yêu cầu trên ta có các vùng:

* nhỏ hơn 1 : vùng không hợp lệ
* [1-10] : vùng hợp lệ

lớn hơn 10 và nhỏ hơn 20 : vùng không hợp lệ

* [20-30] : vùng hợp lệ
* lớn hơn 30: vùng không hợp lệ
* Nhập các ký tự không phải số nguyên : vùng không hợp lệ

+ Vì vậy có các case:

+ Case hợp lệ:

* Nhập 5 ký tự
* Nhập 25 ký tự

+ Case không hợp lệ:

* Không nhập vào trường
* Nhập số thập phân
* Nhập 35 ký tự
* Nhập ký tự chữ: Tiếng việt, Tiếng anh, Full-size, Half-size
* Nhập ký tự đặc biệt, space, kí tự Enter
* Nhập câu lệnh SQL injection, HTML, XSS
* **Phân tích giá trị biên:**

+ Phân tích giá trị biên dựa trên việc kiểm thử tại các ranh giới giữa các phân vùng, Chúng ta sẽ tập trung vào các giá trị biên chứ không test toàn bộ dữ liệu. Thay vì chọn nhiều giá trị trong lớp đương tương để làm đại diện, phân tích giá trị biên yêu cầu chọn một hoặc vài giá trị là các cạnh của lớp tương đương để làm điều kiện test.

+ Chúng ta thường thấy rằng một số lượng lớn lỗi xảy ra tại các ranh giới của các giá trị đầu vào được xác định thay vì các giá trị giữa, còn được gọi là các giá trị biên. Từ đó đưa ra lựa chọn các test cases thực hiện giá trị đầu vào các giá trị biên.

+ Kỹ thuật thiết kế test cases này bổ sung cho phân vùng tương đương. Kỹ thuật kiểm thử phần mềm này dựa trên nguyên tắc: Nếu một hệ thống hoạt động tốt với các giá trị biên thì nó sẽ hoạt động tốt cho tất cả các giá trị nằm giữa hai giá trị biên.

+ Phân tích giá trị biên sẽ chọn các giá trị:

* Giá trị ngay dưới giá trị nhỏ nhất
* Giá trị nhỏ nhất
* Giá trị ngay trên giá trị nhỏ nhất
* Giá trị ngay dưới giá trị lớn nhất
* Giá trị lớn nhất
* Giá trị ngay trên giá trị lớn nhất
* **Bảng quyết định:**

+ Kỹ thuật kiểm thử phần mềm này được sử dụng cho các chức năng cần sự kết hợp của các yếu tố đầu vào các biến.

**+** Ví dụ: Nút Submit phải được enable nếu người dùng đã nhập tất cả các trường bắt buộc.

+ Đầu tiên là xác định đấu ra của các chức năng có phụ thuộc vào sự kết hợp của các đầu vào. Nếu có tập hợp kết hợp đầu vào lớn, thì hãy chia nó thành các tập hợp nhỏ hơn hữu ích cho việc quản lý bảng quyết định.

+ Đối với mọi chức năng, cần tạo một bảng và liệt kê tất cả các loại kết hợp đầu vào và đầu ra tương ứng. Điều này giúp xác định các điều kiện bị tester bỏ qua.

+ Các bước để tạo bảng quyết định:

* Nhập đầu vào theo hàng
* Nhập tất cả các quy tắc trong cột
* Điền vào bảng với sự kết hợp của đầu vào
* Trong hàng cuối cùng, ghi chú đầu ra so với kết hợp đầu vào.
* **Đoán lỗi:**

+ Đoán lỗi là một kỹ thuật kiểm thử phần mềm dựa trên việc đoán lỗi có thể chiếm ưu thế trong code. Đây là một kỹ thuật dựa trên kinh nghiệm, trong đó nhà phân tích kiểm thử sử dụng kinh nghiệm của mình để đoán phần có vấn đề hoặc có lỗi của ứng dụng kiểm thử.

+ Kỹ thuật xác định danh sách các lỗi có thể xảy ra hoặc các tình huống dễ xảy ra lỗi. Sau đó, người kiểm thử viết test cases để tìm kiếm những lỗi đó. Để thiết kế các test cases dựa trên kỹ thuật kiểm thử phần mềm này, nhà phân tích có thể sử dụng các kinh nghiệm trong quá khứ để xác định các điều kiện.

+ Cách đoán lỗi:

* Tester nên sử dụng kinh nghiệm trước đây để kiểm thử các ứng dụng tương tự
* Hiểu biết về hệ thống đang kiểm thử
* Kiến thức về các lỗi thực hiện điển hình
* Nhớ những chức năng phức tạp trước đây
* Đánh giá lịch sử dữ liệu và kết quả kiểm thử

1. Giới thiệu phần mềm
   1. Yêu cầu đề tài

* Trong phần này, nhóm sẽ sử dụng công cụ Selenium cho website TOPCV.VN.
* Giới thiệu về Selenium: Selenium là một trong những công cụ kiểm thử phần mềm tự động mã nguồn mở (open source test automation tool) mạnh mẽ nhất hiện nay cho việc kiểm thử ứng dụng Web. Selenium script có thể chạy được trên hầu hết các trình duyệt như IE, Mozilla FireFox, Chrome, Safari, Opera; và hầu hết các hệ điều hành như Windows, Mac, Linux.
* Selenium là một công cụ kiểm thử phần mềm tự động, được phát triển bởi ThoughtWorks từ năm 2004 với tên ban đầu là JavaScriptTestRunner. Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium team, một phần của Google và phát triển thành Selenium như hiện nay.
* Về cấu trúc, Selenium có 4 phần:

+ [Selenium IDE](http://vntesters.com/selenium-ide-record-and-playback/)

+ Selenium RC (Selenium 1 – Selenium Remote Control)

+ [Selenium Gird](http://vntesters.com/kiem-thu-tu-dong-selenium-grid/)

+ [Selenium WebDriver](http://vntesters.com/gioi-thieu-selenium-webdriver/) (Selenium 2)

* Về đặc điểm:

+ Mã nguồn mở. Phải nói điểm này là điểm mạnh nhất của Selenium khi so sánh với các test tool khác. Vì là mã nguồn mở nên chúng ta có thể sử dụng mà không phải lo lắng về phí bản quyền hay thời hạn sử dụng.

+ Selenium hỗ trợ nhiều ngôn ngữ lập trình.

+ Selenium hỗ trợ chạy trên nhiều OS khác nhau với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của chúng ta.

+ Chạy test case ở background. Khi chúng ta thực thi một test scrpit, chúng ta hoàn toàn có thể làm việc khác trên cùng một PC. Điều này hỗ trợ chúng ta không cần tốn quá nhiều tài nguyên máy móc khi chạy test script.

* Trong webiste này, có rất nhiều chức năng cũng như giao diện để chúng ta có thể kiểm thử, ở đấy nhóm sẽ chọn ra 5 chức năng chính để thực hiện kiểm thử đó là: đăng nhập, đăng ký, ứng tuyển, cập nhật thông tin cá nhân, và cuối cùng là tìm kiếm công việc
* Cài đặt công cụ Selenium IDE:

+) Bước 1: Vào trình duyệt Chrome và truy cập vào link <https://www.selenium.dev/selenium-ide/>

A white background with black and white clouds

Description automatically generated

+) Bước 2: Nhấn thêm vào Chrome

+) Bước 3: Sau khi thêm thành công sẽ hiển thị biểu tượng Selenium IDE.

* Cách sử dụng:

+) Bước 1: Sau khi nhấn vào biểu tượng Selenium IDE. Một cửa sổ được hiển thị lên.

+) Bước 2: Chọn create a new project

A screenshot of a computer

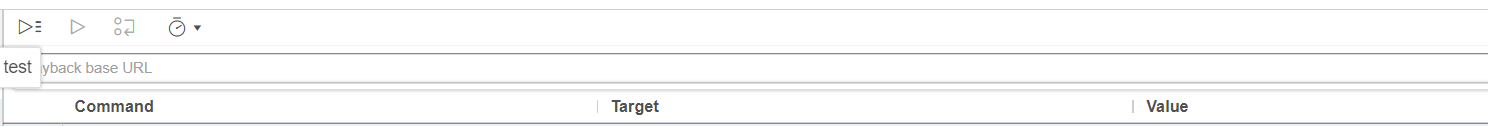
Description automatically generated

+) Bước 3: Nhập tên dự án và nhấn OK.

A screenshot of a computer

Description automatically generated

+) Bước 4: Điền base URL của dự án vào ô “Playback base URL” trên thanh công cụ.



+) Bước 5: Để tạo test case nhấn dấu “+” bên trên thanh công cụ. Điền tên test case và nhấn ADD.

+) Bước 6: Để tạo các hoạt động cho test case bằng cách chọn những Command, target, value tương ứng hoặc thao tác mẫu trên website bằng cách nhấn nút “REC” trên thanh công cụ.

+) Bước 7: Sau khi thao tác mẫu xong thì nhấn nút dừng ghi để dừng quá trình ghi thao tác.

+) Bước 8: Để thực thi test case đang chọn bằng cách chọn test case và nhấn nút Run current test hoặc thực thi tất cả các test bằng cách nhấn Run all tests.

+) Bước 9: Log là nơi ghi chú lại các test, kết quả từng bước và trạng thái cho các test case đã chạy.

* 1. Phân tích thiết kế hệ thống

2.2.1. Phân tích use case Đăng ký

**2.2.1.1. Luồng cơ bản**

1**.** Use case bắt đầu khi người dùng kích vào nút “ Đăng ký” ở trang chủ. Hệ thống sẽ hiển thị form Đăng ký lên màn hình.

2. Người dùng nhập đầy đủ thông tin theo hướng dẫn, sau đó kích vào nút “Đăng ký”. Hệ thống sẽ kiểm tra và lưu thông tin người dùng.

- Use case kết thúc.

**2.2.1.2. Luồng rẽ nhánh**

- Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

- Tại bước 2 trong luồng cơ bản, nếu người dùng không nhập đầy đủ thông tin, hệ thống thông báo thiếu thông tin cần nhập và yêu cầu người dùng nhập lại

- Tại bước 2 trong luồng cơ bản, nếu người dùng nhập sai mật khẩu xác nhận thì hệ thống sẽ đưa ra thông báo “Mật khẩu xác nhận chưa đúng” và yêu cầu khách hàng nhập lại.

**2.2.1.3. Điểm mở rộng**

- Không có

**2.2.1.4. Tiền điều kiện**

- Không có

**2.2.1.5. Hậu điều kiện**

**-** Không có

**2.2.1.6. Yêu cầu đặc biệt**

**-** Không có

**2.2.1.7. Hình dung màn hình**

**A screenshot of a computer

Description automatically generated**

*Hình 4. Trang đăng ký*

2.2.2. Phân tích use case Đăng nhập

**2.2.1.1. Luồng cơ bản**

1. Use case bắt đầu khi người dùng kích vào nút “ Đăng nhập” ở trang chủ. Hệ thống sẽ hiển thị form Đăng nhập lên màn hình.

2. Người dùng nhập đầy đủ thông tin theo hướng dẫn, sau đó kích vào nút “Đăng nhập”. Hệ thống sẽ kiểm tra và đăng nhập vào tài khoản người dùng.

- Use case kết thúc

**2.2.1.2. Luồng rẽ nhánh**

- Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

- Tại bước 2 trong luồng cơ bản, nếu người dùng không nhập đầy đủ thông tin, hệ thống thông báo thiếu thông tin cần nhập và yêu cầu người dùng nhập lại

- Tại bước 2 trong luồng cơ bản, nếu người dùng nhập sai mật khẩu hoặc email thì hệ thống sẽ đưa ra thông báo “Địa chỉ email hoặc mật khẩu chưa đúng” và yêu cầu khách hàng nhập lại.

**2.2.1.3. Điểm mở rộng**

- Không có

**2.2.1.4. Tiền điều kiện**

- Người dùng đã có tài khoản topcv

**2.2.1.5. Hậu điều kiện**

**-** Không có

**2.2.1.6. Yêu cầu đặc biệt**

**-** Không có

**2.2.1.7. Hình dung màn hình**

**A screenshot of a phone

Description automatically generated**

*Hình 5. Trang đăng nhập*

2.2.3. Phân tích use case Tìm kiếm công việc

- Use case này cho phép người dùng tìm kiếm công việc phù hợp.

**2.2.3.1. Luồng cơ bản**

1**.** Use case này bắt đầu khi khách hàng kích vào thanh nhập tìm kiếm ở giao diện trang chủ. Khách hàng nhập tên công việc, sau đó chọn lần lượt nơi làm việc, số năm kinh nghiệm và mức lương đề xuất, ấn vào nút “Tìm kiếm” để thực hiện quá trình tìm kiếm.

2. Hệ thống xử lý thông tin tìm kiếm, đối chiếu thông tin tên công việc trong bảng CONG\_VIEC, thông tin về nơi làm việc, số năm kinh nghiệm hay mức lương đề xuất trong bảng TUY\_CHON\_NANG\_CAO, và hiển thị thông tin công việc cần tìm lên giao diện với các thông tin bao gồm: tên công việc, mô tả, logo công ty, tên công ty, mức lương, nơi làm việc được lấy trong bảng CHI\_TIET\_CONG\_VIEC.

**2.2.3.2. Luồng rẽ nhánh**

- Tại bước 2 trong luồng cơ bản, nếu hệ thống đối chiếu thông tin thấy một trong những yêu cầu như: tìm kiếm, các tùy chọn nâng cao không tồn tại, hệ thống sẽ hiển thị thông báo “Không tìm thấy kết quả nào. Hãy thử lại.” và quay lại bước 1.

- Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiên thị một thông báo lỗi và use case kết thúc.

**2.2.3.3. Điểm mở rộng**

- Tìm kiếm công việc

**2.2.3.4. Tiền điều kiện**

- Người dùng đã có tài khoản topcv

**2.2.3.5. Hậu điều kiện**

**-** Không có

**2.2.3.6. Yêu cầu đặc biệt**

**-** Không có

**2.2.3.7. Hình dung màn hình**

**A screenshot of a computer

Description automatically generated**

*Hình 6. Trang tìm kiếm công việc*

2.2.4. Phân tích use case Cập nhật thông tin cá nhân

- Use case này cho phép người dùng cập nhật thông tin cá nhân.

**2.2.4.1. Luồng cơ bản**

1. Use case này bắt đầu khi người dùng kích vào “A person in a circle

Description automatically generated”, sau đó chọn vào mục “Cài đặt thông tin cá nhân”, lúc này hệ thống sẽ hiển thị ra form cập nhật thông tin cá nhân, gồm họ và tên, số điện thoại và Email được lấy trong bảng TAI\_KHOAN.

2. Người dùng sẽ thay đổi thông tin họ và tên, số điện thoại rồi sau đó ấn nút “Lưu”, hệ thống sẽ kiểm tra tính hợp lệ, tự động lưu lại và tải lại trang.

**2.2.4.2. Luồng rẽ nhánh**

- Tại bước 2 trong luồng cơ bản, thông tin bắt buộc phải có là họ và tên, nếu không nhập họ và tên, hệ thống sẽ thông báo “Họ và tên không được để trống” và quay lại bước 2.

- Tại bước 2 trong luồng cơ bản, thông tin không bắt buộc là số điện thoại, có thể nhập hoặc không nhập, thông tin Email sẽ mặc định là email ban đầu, không thể thay đ

- Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiên thị một thông báo lỗi và use case kết thúc.

**2.2.4.3. Điểm mở rộng**

- Cập nhật thông tin cá nhân

**2.2.4.4. Tiền điều kiện**

- Cần đăng nhập vào tài khoản để sử dụng chức năng này.

**2.2.4.5. Hậu điều kiện**

**-** Các thông tin của người dùng được lưu lại sẽ được cập nhật trong cơ sở dữ liệu.

**2.2.4.6. Yêu cầu đặc biệt**

**-** Không có

**2.2.4.7. Hình dung màn hình**

A screenshot of a computer

Description automatically generated

*Hình 7. Trang cập nhật thông tin cá nhân*

2.2.5. Phân tích use case Ứng tuyển

- Use case này cho phép người dùng đăng ký ứng tuyển cho công việc mình lựa chọn.

**2.2.5.1. Luồng cơ bản**

1. Use case này bắt đầu khi người dùng kích vào nút “Ứng tuyển ngay” ở trang chi tiết tuyển dụng. Hệ thống sẽ hiển thị cửa sổ form đăng ký ứng tuyển bao gồm file CV, họ và tên, email, số điện thoại, thư giới thiệu. Người dùng có thể chọn thông tin CV đã có sẵn trên hệ thống hoặc tải lên CV mới.

2. Sau khi điền đầy đủ thông tin, người dùng kích vào nút “Nộp hồ sơ ứng tuyển”. Hệ thống sẽ tạo mới hồ sơ ứng tuyển trong bảng UNGTUYEN và thông báo thành công cho người dùng.

**2.2.5.2. Luồng rẽ nhánh**

- Tại bước 1 ở luồng cơ bản, nếu người dùng chưa đăng nhập, hệ thống sẽ hiển thị cửa sổ yêu cầu đăng nhập hoặc đăng ký.

- Tại bước 2 ở luồng cơ bản, nếu người dùng đã đăng nhập nhưng chưa xác minh địa chỉ email, hệ thống sẽ hiển thị cửa sổ yêu cầu người dùng xác minh địa chỉ email trước khi nộp hồ sơ ứng tuyển

- Tại bước 2 ở luồng cơ bản nếu người dùng không điền đầy đủ thông tin yêu cầu hoặc sai yêu cầu về dữ liệu thì hệ thống sẽ báo điền các thông tin còn thiếu và các thông tin sai dữ liệu, người dùng không thể nộp hồ sơ.

- Tại thời điểm bất kỳ nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống hiển thị 1 thông báo lỗi và use case kết thúc.

**2.2.5.3. Điểm mở rộng**

- Đăng nhập

**2.2.5.4. Tiền điều kiện**

- Cần đăng nhập vào tài khoản để sử dụng chức năng này.

**2.2.5.5. Hậu điều kiện**

**-** Nếu use case thành công, hồ sơ ứng tuyển sẽ được tạo và thông tin của hồ sơ sẽ được cập nhật trong cơ sở dữ liệu.

**2.2.5.6. Yêu cầu đặc biệt**

**-** Không có

A screenshot of a computer

Description automatically generated

*Hình 8. Trang ứng tuyển*

1. Kiểm thử phần mềm
   1. Lập kế hoạch

#### Giới thiệu chung về dự án

Website topcv.vn là một hệ thống website chuyên về việc tìm việc làm, tra cứu thông tin tuyển dụng, cũng như viết CV trực tuyến giúp người dùng có thể tìm được công việc trong ngành IT, gửi CV online và có cho mình công việc phù hợp

#### Phạm vi kiểm thử

* Các chức năng được kiểm thử:
* Đăng nhập: Kiểm tra chức năng đăng nhập vào hệ thống với tài khoản vừa tạo.
* Đăng ký: Kiểm tra chức năng đăng ký vào hệ thống
* Tìm kiếm công việc
* Cập nhật thông tin cá nhân
* Ứng tuyển
* Các giao diện được kiểm thử:
  + Đăng nhập
  + Đăng ký
  + Trang chi tiết công việc
  + Xem hồ sơ cá nhân
  + Tìm kiếm công việc
  + Trang ứng tuyển công việc
* Các rủi ro có thể xảy ra:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Rủi ro** | **Cách khắc phục** | **Mức độ rủi ro** |
| 1 | Thay đổi yêu cầu làm ảnh hưởng đến nguồn nhân lực và chiến lược test. | Lập lại plan sao cho phù hợp với lịch trình thực tế khi thay đổi yêu cầu, có thể chọn cách tăng thêm nguồn nhân lực cho dự án, hoặc tăng thời gian làm việc ngoài giờ cho nhân viên. | Cao |
| 2 | Sản phẩm mà lập trình viên thực hiện không kịp theo thời gian như lịch trình đề ra. | Yêu cầu cập nhật tiến độ công việc thường xuyên để quản lý kịp thời các thay đổi về thời gian và kỹ thuật. | Cao |
| 3 | Lỗi trong thiết kế test case Kiểm tra lại yêu cầu, kết quả và đầu vào. | Kiểm tra lại yêu cầu, kết quả và đầu vào. | Trung bình |

#### Các tiêu chí chấp nhận sản phẩm

* Tỉ lệ test case đạt( passed): 100%
* Tỉ lệ test case không đạt (failed): 0%
* Hệ thống chạy ổn định trên các trình duyệt web khác nhau (IE, Firefox và Google Chrome, Microsoft Edge).

#### Các tiêu chí chấp nhận sản phẩm

* Kiểm thử ở mức hệ thống (ST) và kiểm thử chấp nhận (UAT)
* Dùng kiểu kiểm thử thủ công (manual test) bao gồm kiểm thử GUI và từng chức năng.
* Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử GUI và chức năng.
* Thiết kế test case theo phương pháp phân vùng tương đương.
* Chỉ thực hiện kiểm thử hồi quy, không thực hiện kiểm thử lại.
* Các yêu cầu phi chức năng khác: tải trọng, hiệu năng…không được kiểm thử

#### Lịch trình công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mốc công việc** | **Sản phẩm** | **Thời gian** | **Bắt đầu** | **Kết thúc** |
| Lập kế hoạch kiểm thử | Test plan | 5 ngày | 28/03/2024 | 2/4/2024 |
| Xem lại các tài liệu | Test plan | 4 ngày | 2/4/2024 | 06/04/2024 |
| Thiết kế các testcase | Test case | 16 ngày | 06/04/2024 | 22/04/2024 |
| Viết các testcase | Test case | 11 ngày | 23/04/2024 | 04/05/2024 |
| Xem lại các testcase | Test case | 2 ngày | 04/05/2024 | 06/05/2024 |
| Thực thi các testcase | Test case | 2ngày | 07/05/2024 | 09/05/2024 |
| Ghi nhận và đánh giá kết quả kiểm thử | Test report | 3 ngày | 10/05/2024 | 13/05/2024 |

#### Sản phẩm bàn giao

| **STT** | **Sản phẩm** | **Ngày bàn giao** | **Người bàn giao** | **Người nhận** |
| --- | --- | --- | --- | --- |
| 1 | Test plan | 04/04/2024 | Nhóm 8 | Th.S Nguyễn Thái Cường |
| 2 | Test case | 21/04/2024 | Nhóm 8 | Th.S Nguyễn Thái Cường |
| 3 | Test report | 13/05/2024 | Nhóm 8 | Th.S Nguyễn Thái Cường |

* 1. Thực hiện kiểm thử

3.2.1 Lê Tuấn Hưng-Đăng nhập

***- Phân tích thiết kế kiểm thử***

* Giao diện đăng nhập:

A screenshot of a computer

Description automatically generated

*Hình 10. Trang đăng nhập*

* Lưu đồ thuật toán:

A diagram of a flowchart

Description automatically generated

*Hình 11. Lưu đồ thuật toán*

* Thiết kế test case:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đầu vào | Vùng hợp lệ | Ký hiệu đánh dấu | Vùng không hợp lệ | Đánh dấu |
| Email | (5-20) ký tự | v | Dưới 5 ký tự | ✗ Độ dài không hợp lệ |
|  |  |  | Trên 20 ký tự | ✗ Độ dài không hợp lệ |
|  | Ký tự chữ cái hoặc số | v | Ký tự đặc biệt và khoảng trắng | ✗ Ký tự không hợp lệ |
| Mật khẩu | Độ dài (8-16) ký tự | v | Độ dài < 8 hoặc > 16 ký tự | ✗ Độ dài không hợp lệ |
|  | Chữ cái, số, ký tự đặc biệt | v | Chứa khoảng trắng hoặc ký tự không in được | ✗ Ký tự không hợp lệ |

* Danh sách test case:

A screenshot of a computer

Description automatically generated

*Hình 12. Test case đăng nhập*

***- Thực hiện kiểm thử***

Bước 1: Tạo test case mới:

A screenshot of a computer

Description automatically generated

*Hình 13. Tạo test case mới*

Bước 2: Nhập đường link trang web cần test

A white line with black dots

Description automatically generated with medium confidence

*Hình 14. Trang thêm link URL*

Bước 3: Thực hiện thao tác test case

A screenshot of a computer

Description automatically generated

*Hình 15. Danh sách thao tác*

Bước 4: Run lại test hệ thống thông báo kiểm thử thành công

A screenshot of a computer

Description automatically generated

*Hình 16. Test case thành công*

**=>** Tỉ lệ test case đạt (passed): 100%

3.2.2 Trần Tiến Thịnh - Tìm kiếm công việc

***- Phân tích thiết kế kiểm thử***

* Giao diện tìm kiếm công việc:



*Hình 17. Giao diện tìm kiếm công việc*

Mô tả giao diện và các điều kiện ràng buộc:

+ Giao diện tìm kiếm gồm Text Input và Button, và bộ lọc dựa trên các tiêu chí: vị trí, kinh nghiệm, lương. Người dùng nhập tên công việc muốn tìm vào TextInput và kích biểu tượng “Tìm kiếm”

+ Điều kiện ràng buộc: từ khóa không được để trống và tối đa 255 ký tự

- Sử dụng các phương pháp kỹ thuật đã học để lấy cơ sở tìm TC

* Lưu đồ thuật toán:

A diagram of a flowchart

Description automatically generated

*Hình 18. Lưu đồ thuật toán*

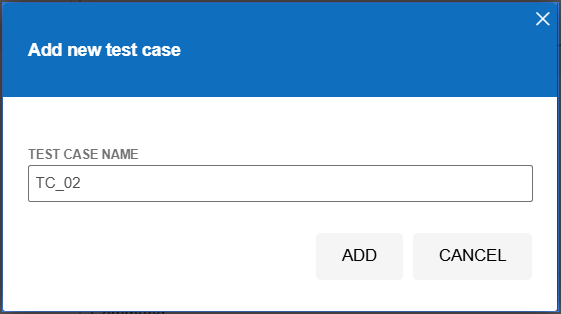
* Thiết kế test case:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đầu vào | Vùng hợp lệ | Ký hiệu đánh dấu | Vùng không hợp lệ | Đánh dấu |
| Email | (5-20) ký tự | v | Dưới 5 ký tự | ✗ Độ dài không hợp lệ |
|  |  |  | Trên 20 ký tự | ✗ Độ dài không hợp lệ |
|  | Ký tự chữ cái hoặc số | v | Ký tự đặc biệt và khoảng trắng | ✗ Ký tự không hợp lệ |
| Mật khẩu | Độ dài (8-16) ký tự | v | Độ dài < 8 hoặc > 16 ký tự | ✗ Độ dài không hợp lệ |
|  | Chữ cái, số, ký tự đặc biệt | v | Chứa khoảng trắng hoặc ký tự không in được | ✗ Ký tự không hợp lệ |

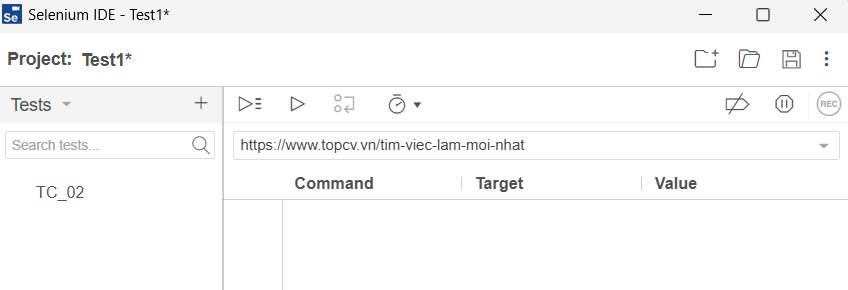
***- Thực hiện kiểm thử***

- Các bước thực hiện kiểm thử tự động : Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE:

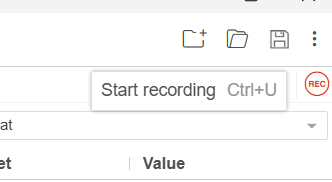
Bước 1: Tạo test case mới:



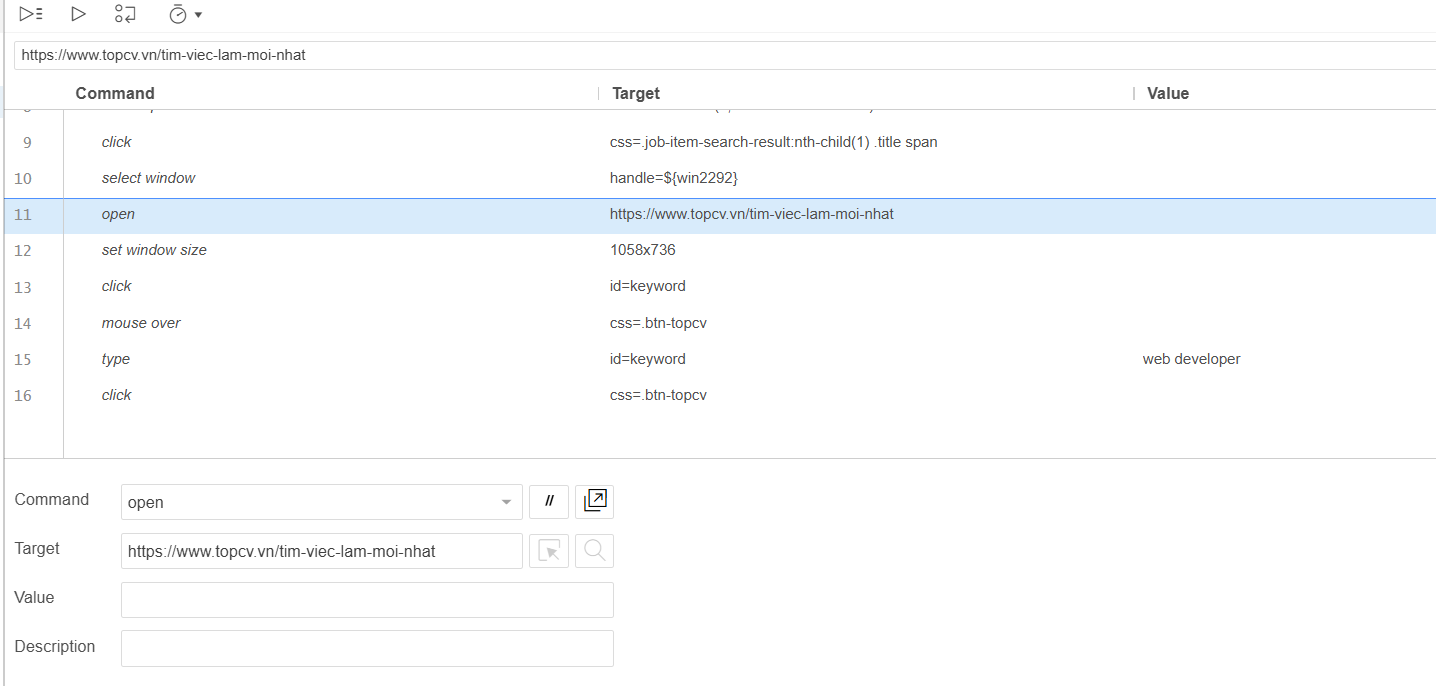
Bước 2: Nhập đường dẫn của trang web:



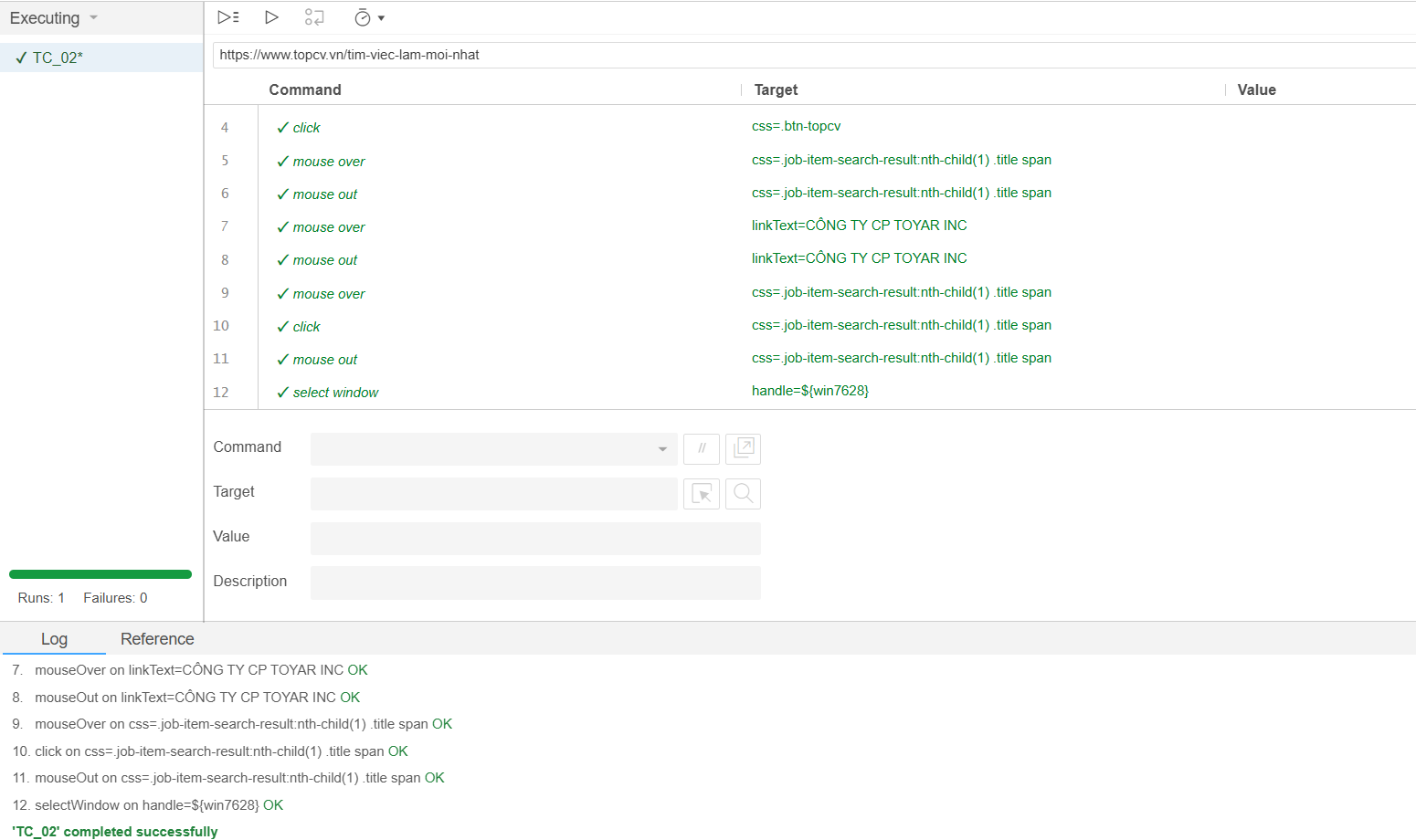
Bước 3: Chọn “REC” để bắt đầu viết kịch bản:



Ta có kịch bản kiểm thử:



Bước 4: Chọn “Run” trong Selenium để tiến hành kiểm thử.



=> Tỉ lệ test case đạt (passed): 100%

3.2.3 Chu Văn Quý - Đăng ký

***- Phân tích thiết kế kiểm thử***

* Giao diện chức năng:

**

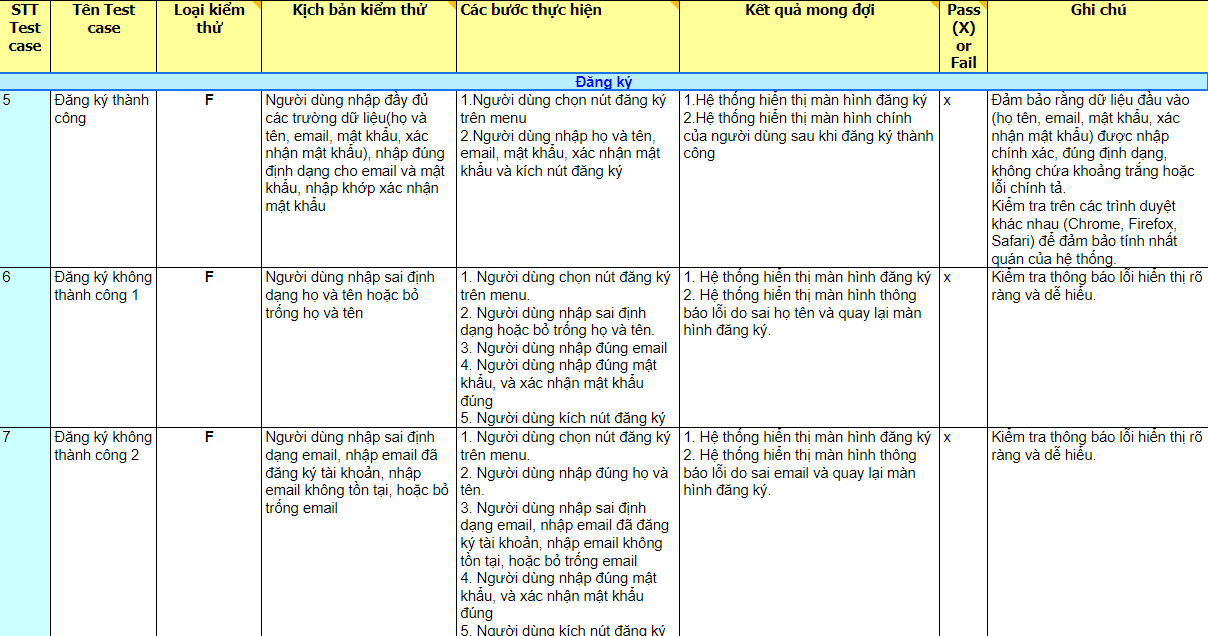
*Hình 19. Giao diện đăng ký*

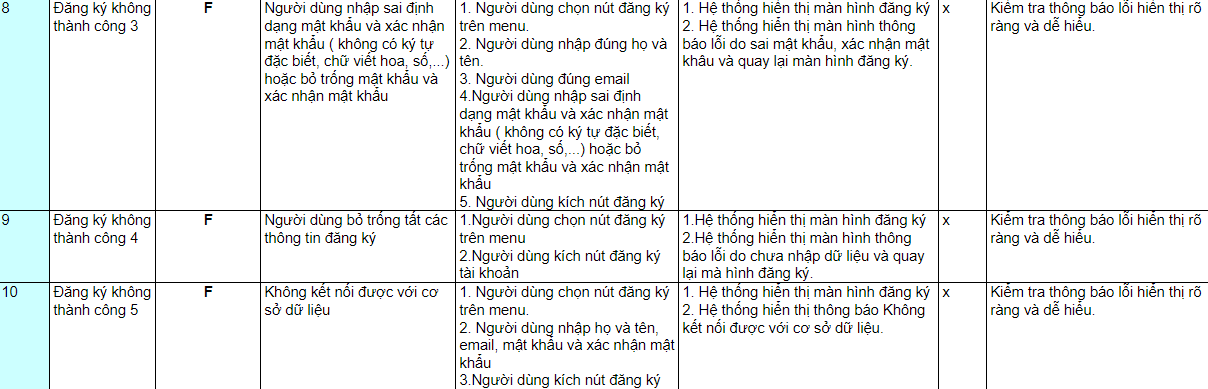
* Mô tả giao diện và các điều kiện ràng buộc:
* Đăng ký bao gồm Họ tên, Email, Mật khẩu, Nhập lại mật khẩu.
* Điều kiện ràng buộc: Họ tên, Email, Mật khẩu, Nhập lại mật khẩu không được để trống và tối đa 255 ký tự.
* Sử dụng các phương pháp kỹ thuật đã học để lấy cơ sở tìm testcase.
* Lưu đồ thuật toán:

\*Sử dụng phương pháp phân hoạch tương đương

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | ký hiệu đánh dấu | Các lớp tương đương không hợp lệ | Đánh dấu |
| Họ tên | 1.> 0 và <= 255 | v | 2.Để trống  3.> 255 | ✗ Độ dài không hợp lệ |
| Email | 4.> 0 và <= 255  5.Nhập đúng định dạng | v | 6.Để trống  7.Nhập sai định dạng | ✗ Độ dài không hợp lệ  ✗ Ký tự không hợp lệ |
| Mật khẩu | 8.> 0 và <= 255  9.Nhập đúng định dạng ( viết hoa, ký tự đặc biệt,...) | v | 10..Để trống  11.Nhập sai định dạng | ✗ Độ dài không hợp lệ  ✗ Ký tự không hợp lệ |
| Nhập lại mật khẩu | 12. Nhập lại đúng mật khẩu | v | 13. Để trống  14. Nhập lại sai mật khẩu | ✗ Ký tự không hợp lệ |

Danh sách test case:



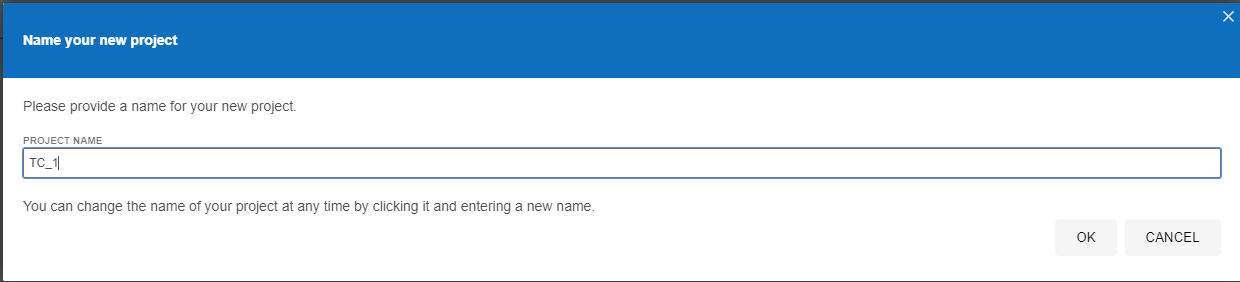


*Hình 20. Danh sách test case*

***- Thực hiện kiểm thử***

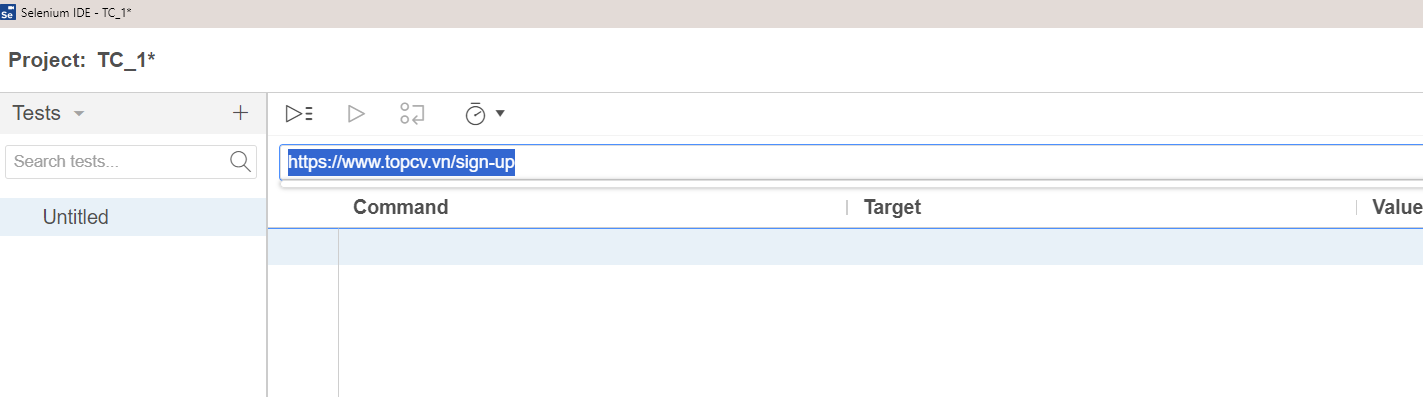
* Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE

Bước 1: Tạo test mới



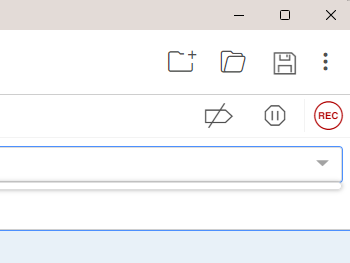
*Hình 21. Giao diện tạo test case mới*

Bước 2: Nhập đường dẫn của trang web

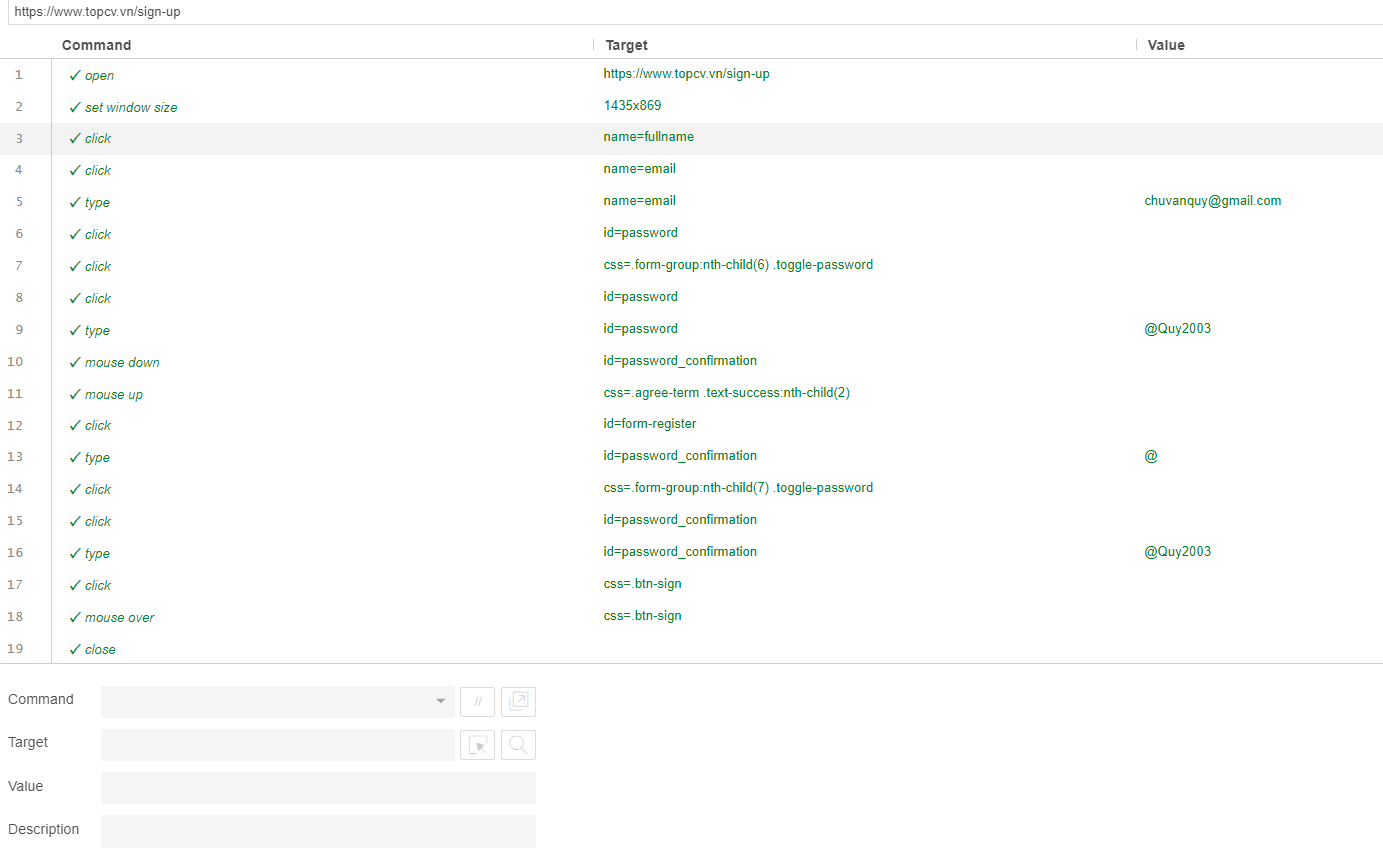


*Hình 22. Trang lưu đường dẫn*

Bước 3: Chọn “REC” để bắt đầu viết kịch bản:

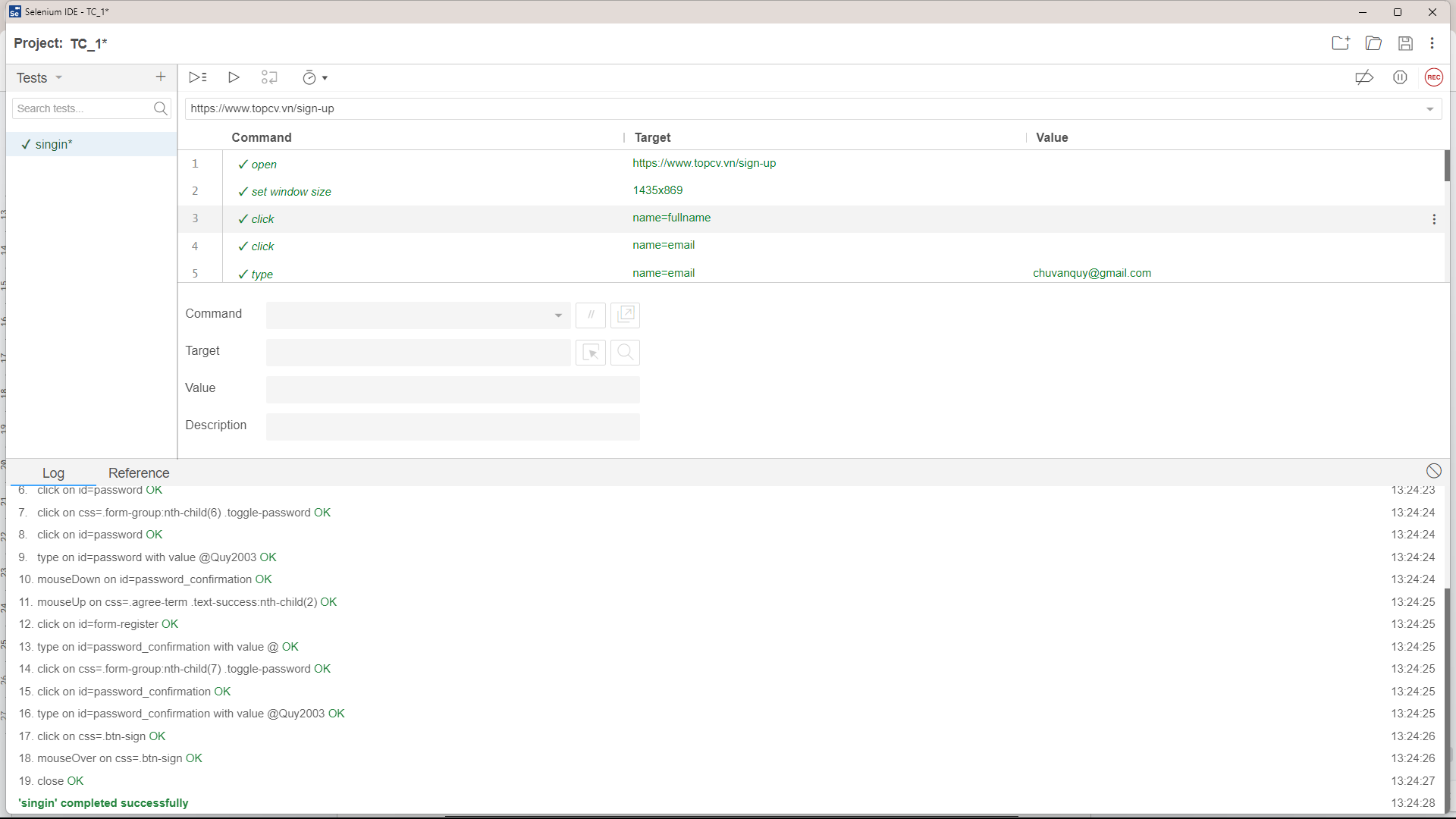


Ta có kịch bản kiểm thử:



*Hình 23. Giao diện danh sách thao tác*

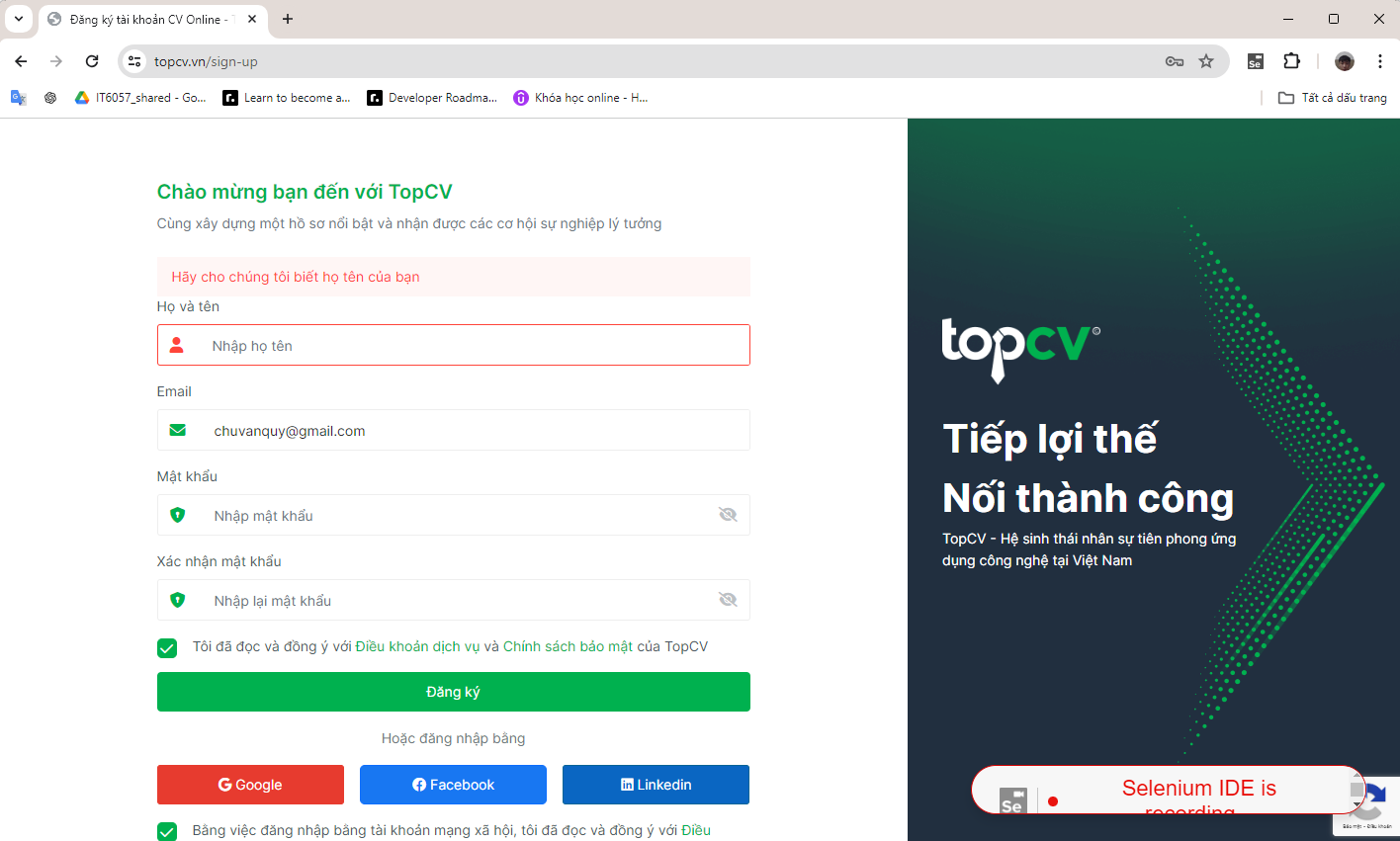
Bước 4: Chọn “Run” trong Selenium để tiến hành kiểm thử.



*Hình 24. Test case thành công*

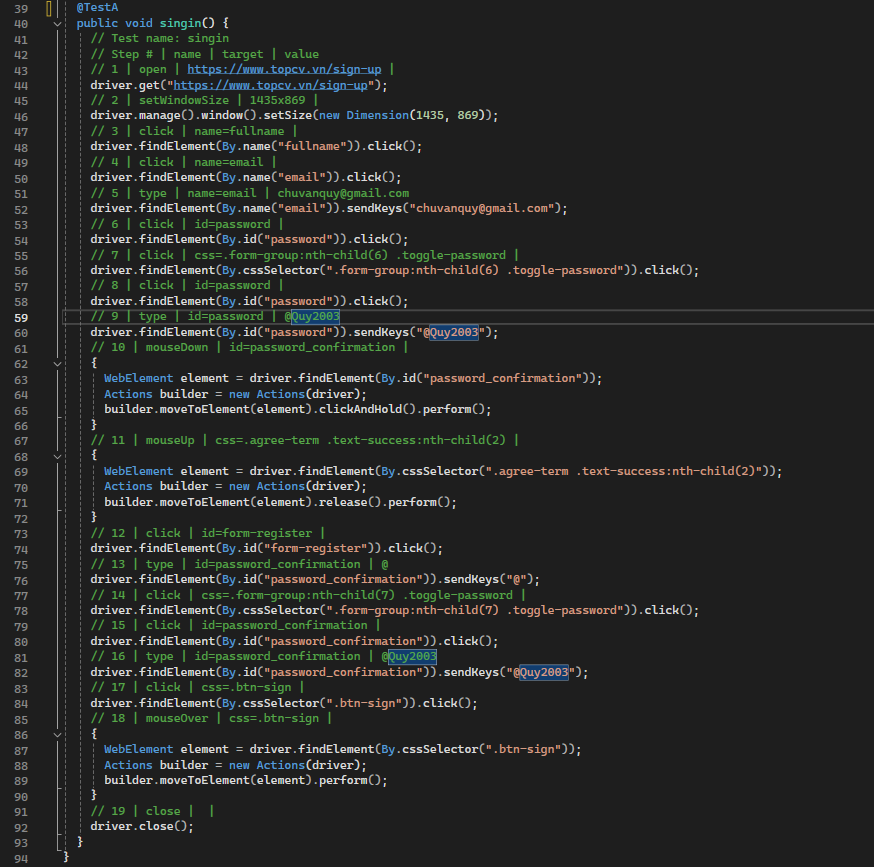
=> Hệ thống thông báo kiểm thử thành công.

* Màn hình kết quả kiểm thử khi để trống họ tên:



* TestScript:





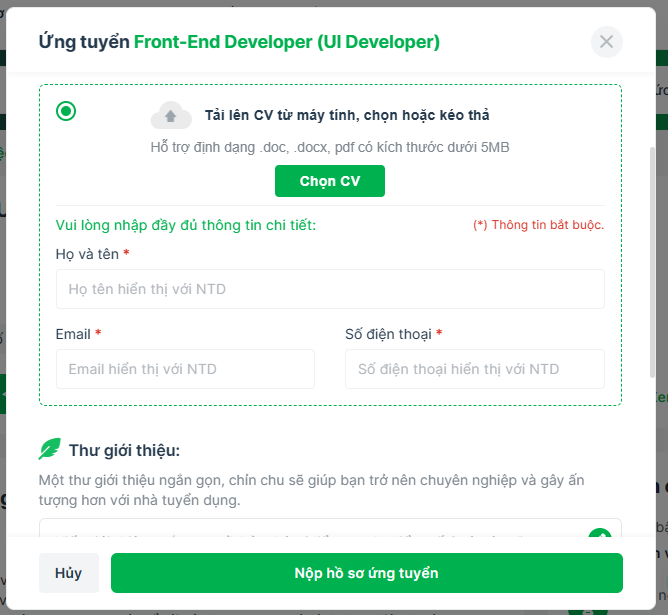
Tương tự lặp lại các bước với các test case trên

=>Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

3.2.4 Ứng tuyển - Nguyễn Dũng Tuấn

***- Phân tích thiết kế kiểm thử***

* Giao diện chức năng:

****

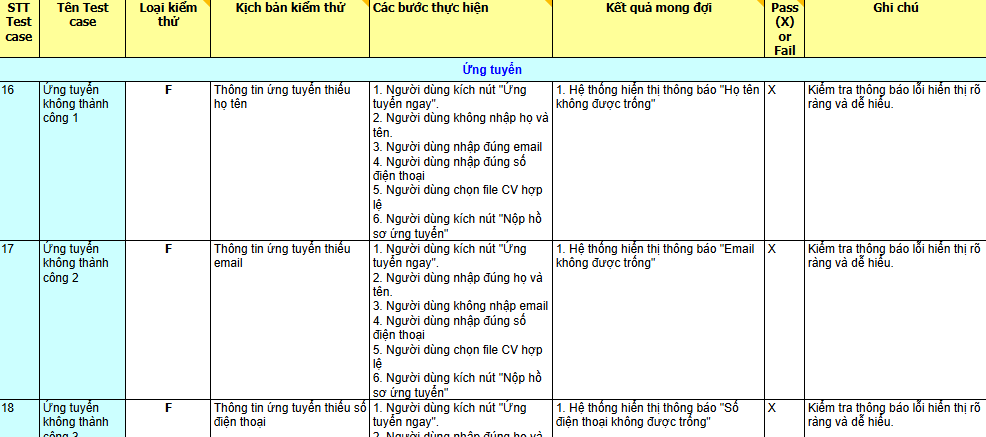
* Mô tả giao diện và các điều kiện ràng buộc:
* Yêu cầu đăng nhập và xác minh email trước khi ứng tuyển
* Ứng tuyển bao gồm Họ tên, Email, Số điện thoại, file CV, thư giới thiệu
* Điều kiện ràng buộc: Họ tên, Email, Số điện thoại không được để trống và tối đa 255 ký tự. File CV không được để trống chỉ hỗ trợ định dạng .doc, .docx, pdf có kích thước dưới 5MB. Thư giới thiệu là không bắt buộc.

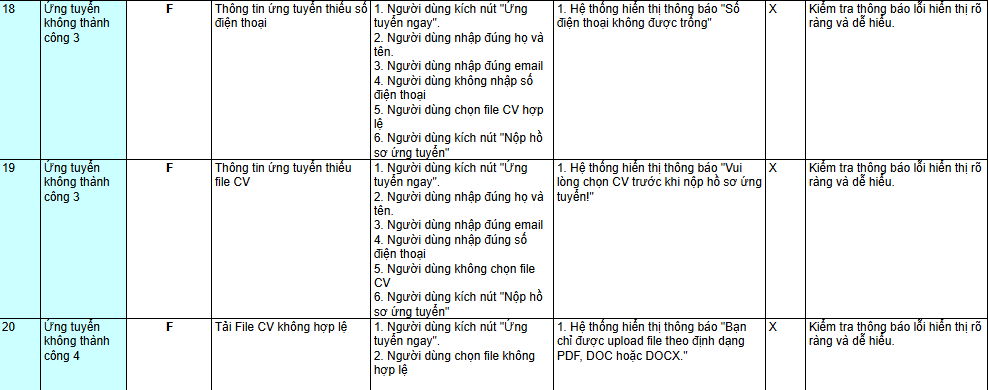
- Sử dụng các phương pháp kỹ thuật đã học để lấy cơ sở tìm TC

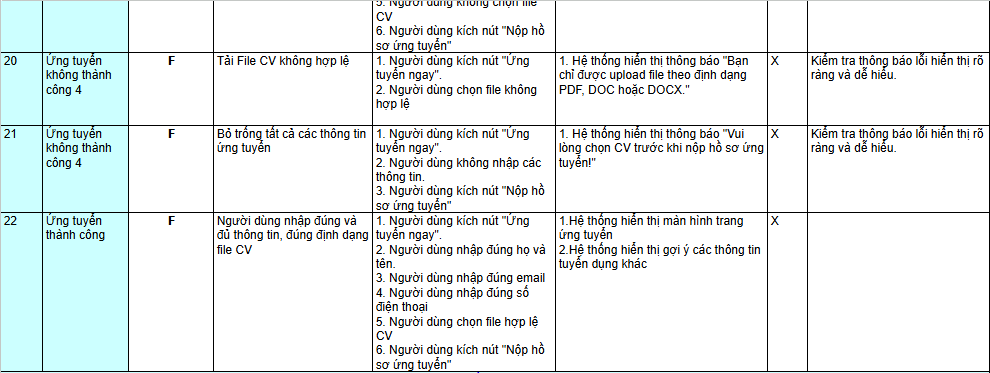
\*Sử dụng phương pháp phân hoạch tương đương

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | ký hiệu đánh dấu | Các lớp tương đương không hợp lệ | Đánh dấu |
| Họ tên | 1.> 0 và <= 255 | v | 2.Để trống  3.> 255 | ✗ Độ dài không hợp lệ |
| Email | 4.> 0 và <= 255 | v | 5.Để trống | ✗ Độ dài không hợp lệ |
| Số điện thoại | 6.> 0 và <= 255 | v | 7.Để trống | ✗ Độ dài không hợp lệ |
| File CV | 8. Chọn đúng loại file | v | 9. Để trống  14. Chọn file không hợp lệ | ✗ Chưa có file  ✗ File không hợp lệ |

Danh sách test case:



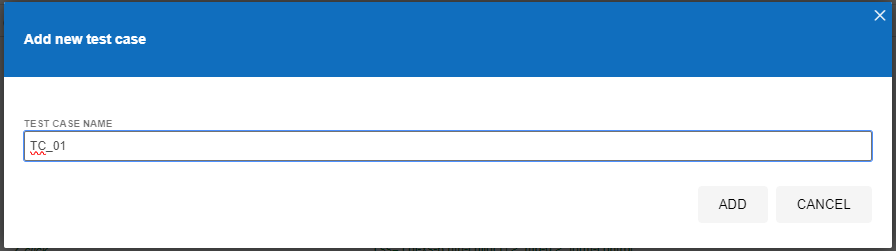




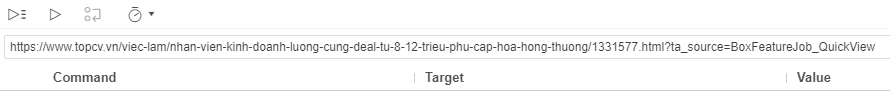
***Thực hiện kiểm thử***

- Các bước thực hiện kiểm thử tự động : Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE

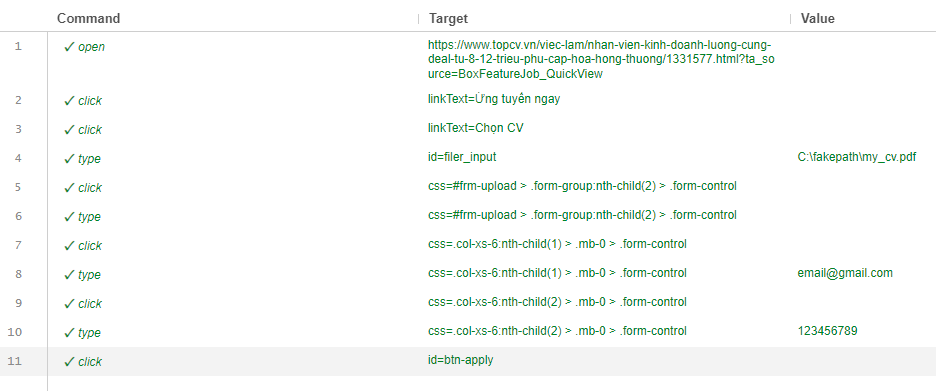
Bước 1: Tạo test mới



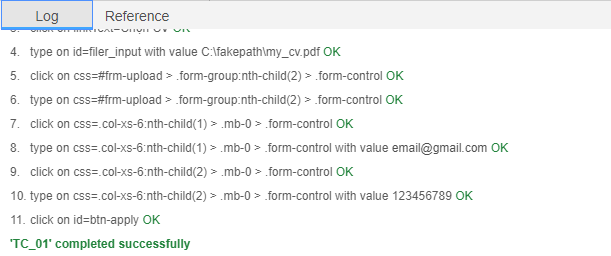
Bước 2: Nhập đường dẫn trang web



Bước 3: Chọn “REC” để bắt đầu viết kịch bản:

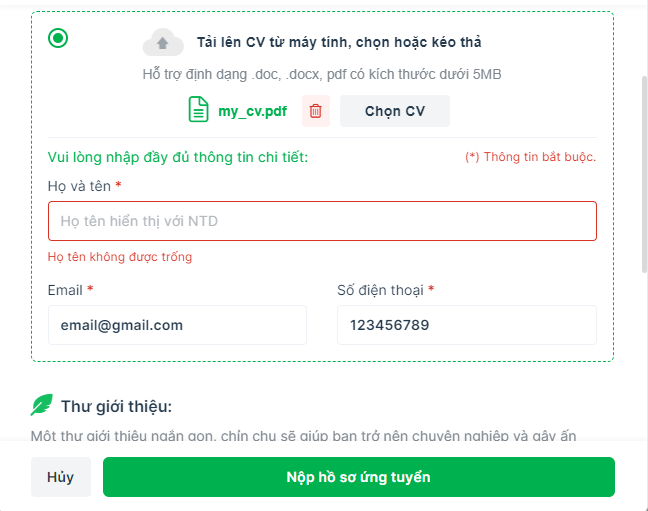


Bước 4: Chọn “Run” trong Selenium để tiến hành kiểm thử.

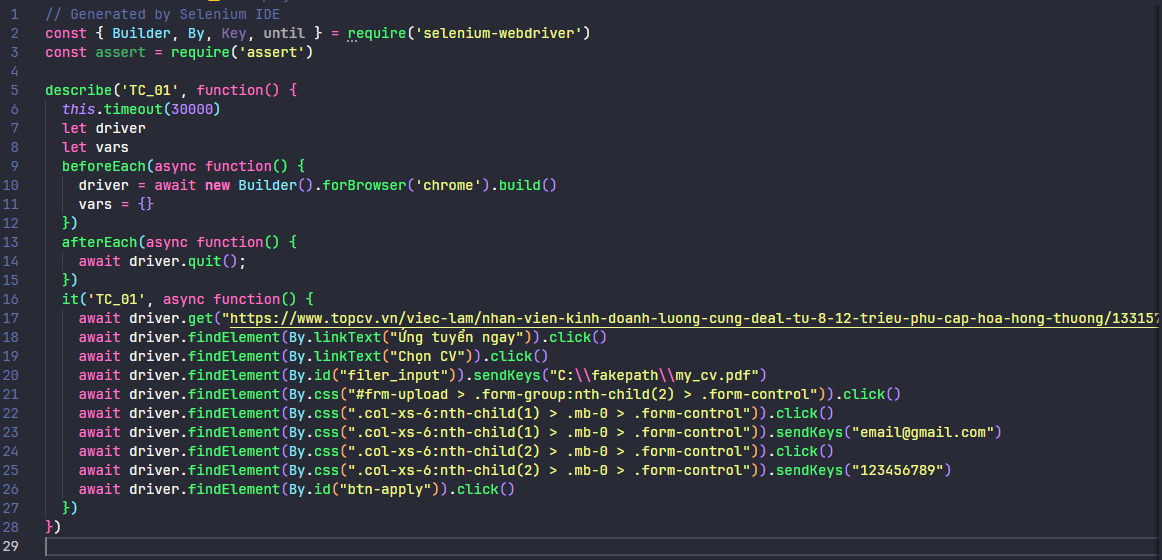


=> Hệ thống thông báo kiểm thử thành công.

* Màn hình kết quả kiểm thử khi để trống họ tên:



TestScript:



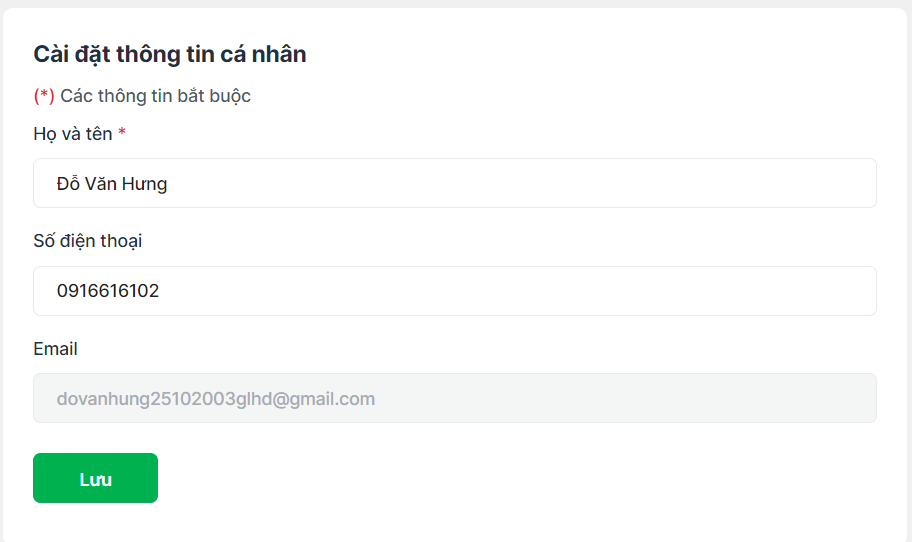
Tương tự lặp lại các bước với các test case trên

=>Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

3.2.5 Cập nhật thông tin cá nhân - Đỗ Văn Hưng

***- Phân tích thiết kế kiểm thử***

* Giao diện chức năng:

****

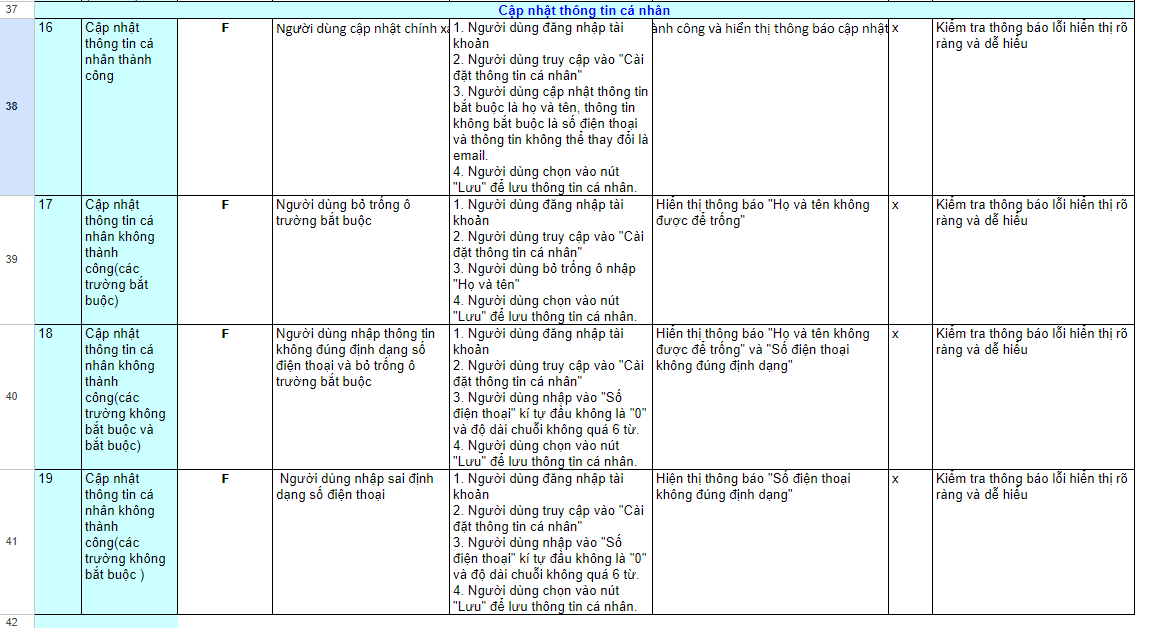
* Mô tả giao diện và các điều kiện ràng buộc:
* Yêu cầu cập nhật thông tin cá nhân hợp lệ
* Các trường bắt buộc là: Họ và tên
* Các trường không bắt buộc là: Số điện thoại
* Các trường không thể thay đổi: Email
* Điều kiện ràng buộc: Số điện thoại phải bắt đầu bằng kí tự “0” và tối thiểu 7 kí tự chữ số.

- Sử dụng các phương pháp kỹ thuật đã học để lấy cơ sở tìm TC

\*Sử dụng phương pháp phân hoạch tương đương

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | ký hiệu đánh dấu | Các lớp tương đương không hợp lệ | Đánh dấu |
| Họ tên | Bất kì kí tự nào | v | Không có | ✗ Không có |
| Email | Không thể thay đổi | v | Không thể thao tác | ✗ Không có |
| Số điện thoại | Bắt đầu từ kí tự “0” và tối thiểu là 7 chữ số | v | Sai định dạng số điện thoại | ✗ Độ dài không hợp lệ hoặc đầu vào không hợp lệ |

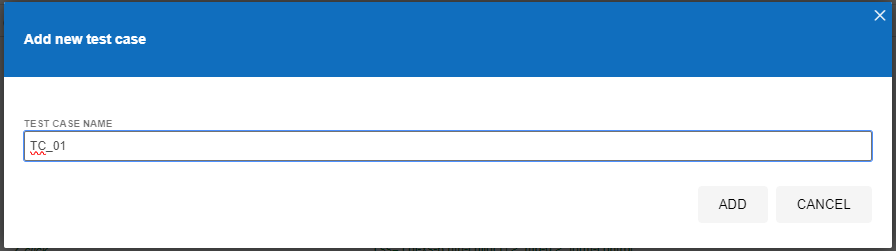
Danh sách test case



***- Thực hiện kiểm thử***

- Các bước thực hiện kiểm thử tự động : Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE

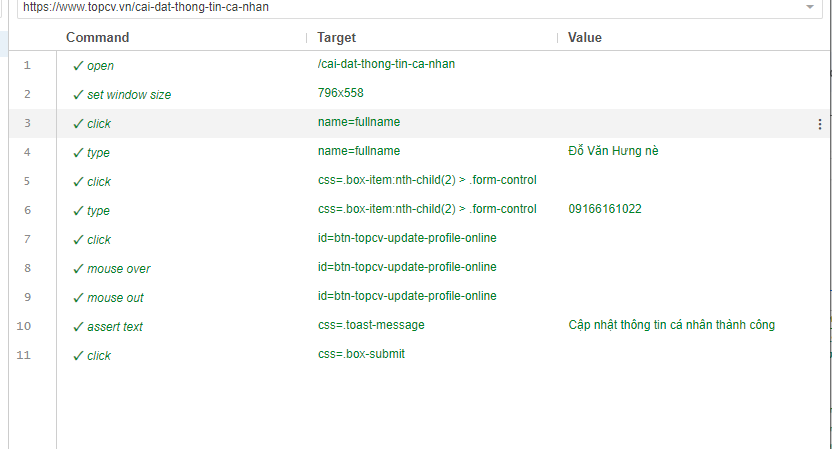
Bước 1: Tạo test mới



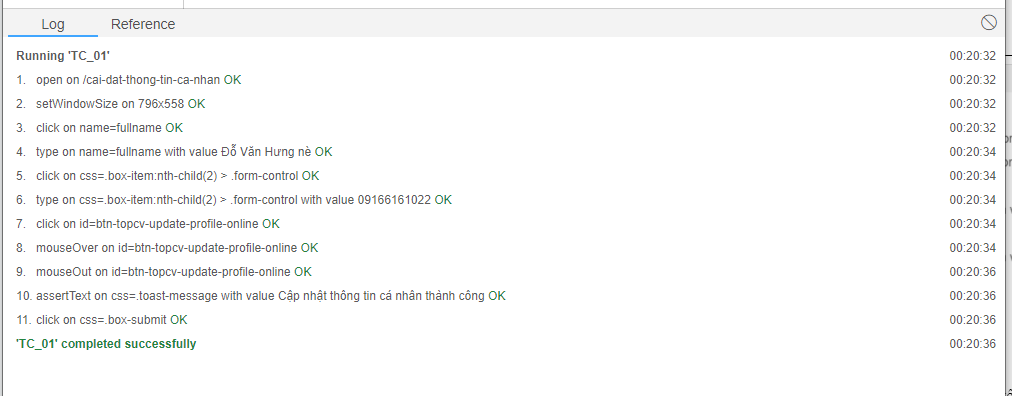
Bước 2: Nhập đường dẫn trang web



Bước 3: Chọn “REC” để bắt đầu viết kịch bản:

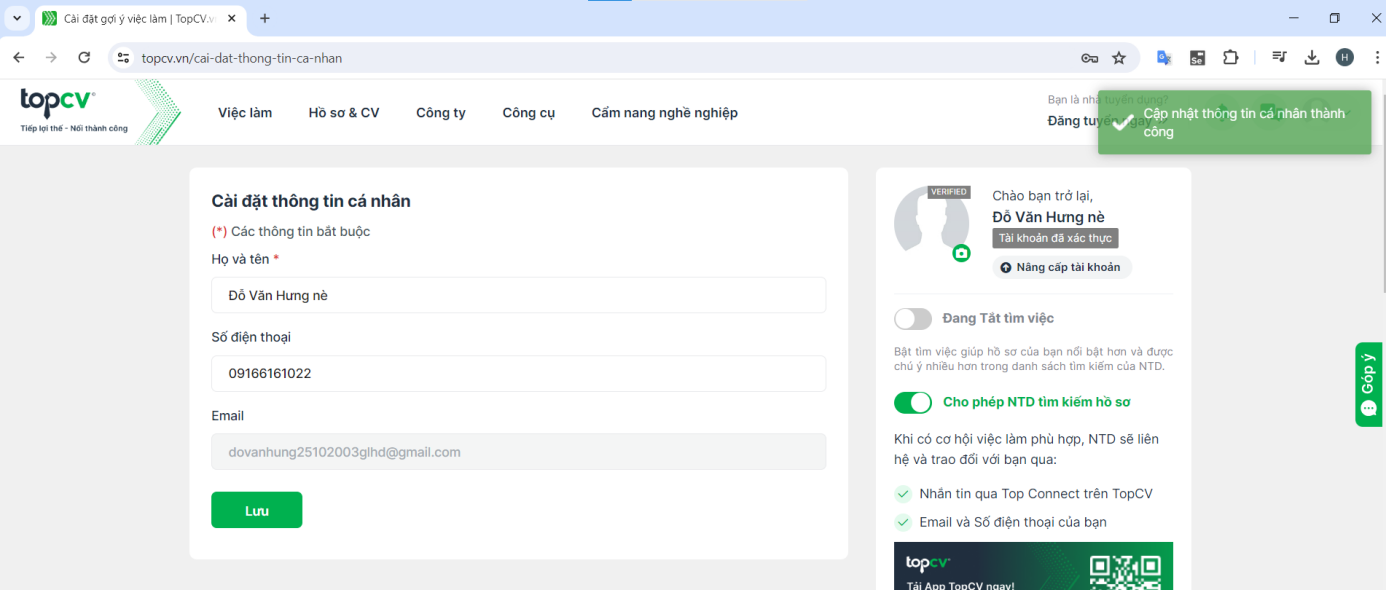


Bước 4: Chọn “Run” trong Selenium để tiến hành kiểm thử.

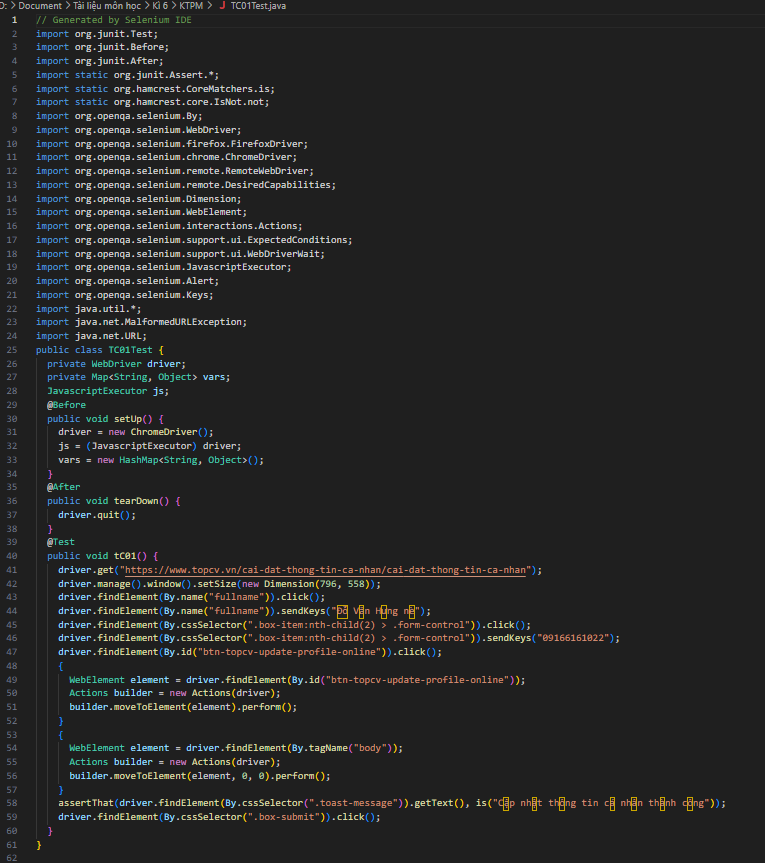


=> Hệ thống thông báo kiểm thử thành công.

* Màn hình kết quả kiểm thử khi để trống họ tên:



TestScript:



Tương tự lặp lại các bước với các test case trên

=>Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

TỔNG KẾT

Nghiên cứu và ứng dụng công cụ Selenium IDE vào kiểm thử website TopCV.vn đã mang lại nhiều kết quả khả quan và hữu ích. Việc sử dụng Selenium IDE không chỉ giúp tăng cường hiệu quả của quá trình kiểm thử mà còn giảm thiểu thời gian và công sức cần thiết cho việc phát hiện và khắc phục lỗi. Selenium IDE cung cấp một giao diện thân thiện và dễ sử dụng, cho phép ngay cả những người không có nền tảng kỹ thuật sâu rộng cũng có thể thực hiện kiểm thử tự động. Điều này đặc biệt quan trọng trong việc đảm bảo chất lượng cho website TopCV.vn. Việc tự động hóa các quy trình kiểm thử không chỉ giúp tiết kiệm thời gian mà còn đảm bảo rằng các bài kiểm thử được thực hiện một cách nhất quán và chính xác, giúp phát hiện sớm các lỗi tiềm ẩn và cải thiện chất lượng sản phẩm cuối cùng. Qua quá trình kiểm thử tự động với Selenium IDE, nhiều lỗi và vấn đề trên website đã được phát hiện và khắc phục kịp thời, góp phần nâng cao trải nghiệm người dùng và độ tin cậy của hệ thống. Từ đó, nghiên cứu này không chỉ minh chứng cho tính hiệu quả của Selenium IDE mà còn mở ra những hướng đi mới cho việc ứng dụng các công cụ kiểm thử tự động trong phát triển và bảo trì các ứng dụng web khác.

Tài liệu tham khảo

* 1. Software testing foundation – Andreas Spilner, Tilo Tinz, Hans Schaefe
  2. Website topcv.vn
  3. **"**Kiểm Thử Tự Động với Selenium**"** -Trần Minh Nhật
  4. **"**Hướng Dẫn Sử Dụng Selenium IDE**"** - Trang kipalog.com