

**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**……** 🙦 🕮 🙤 **……**

**ĐỒ ÁN MÔN HỌC KỸ THUẬT PHẦN MỀM**

***Đề tài***

**KIỂM THỬ TỰ ĐỘNG – KATALON STUDIO**

**Giáo viên hướng dẫn: NGUYỄN THỊ NGỌC**

**Sinh viên thực hiện:**

|  |  |
| --- | --- |
| 1. **TRẦN QUỐC BẢO**   **MSSV: 0306221003** |  |
| **LỚP: CĐ TH 22DĐ D**  **KHÓA: 2022 – 2025** |

**TP. HCM, 25 tháng 3 năm 2024**

**MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU 1**](#_Toc181292771)

[**1.1. TỔNG QUÁT 1**](#_Toc181292772)

[**1.1.1. Khái niệm 1**](#_Toc181292773)

[**1.1.2. Quá trình ra đời 1**](#_Toc181292774)

[**1.1.3. Lý do ra đời 1**](#_Toc181292775)

[**1.1.4. Cách tải 2**](#_Toc181292776)

[**1.2. GIAO DIỆN 3**](#_Toc181292777)

[**1.2.1. Tổng quan giao diện Katalon Studio 3**](#_Toc181292778)

[**1.2.2. Các khái niệm liên quan 8**](#_Toc181292779)

[**1.3. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM 9**](#_Toc181292780)

[**1.3.1. Ưu điểm 9**](#_Toc181292781)

[**1.3.2. Nhược điểm 9**](#_Toc181292782)

[**1.4. CÁCH SỬ DỤNG PHẦN MỀM 10**](#_Toc181292783)

[**1.5. TÍNH ỨNG DỤNG CỦA KATALON STUDIO 12**](#_Toc181292784)

[**1.5.1. Kiểm thử giao diện web (Web UI Testing) 12**](#_Toc181292785)

[**1.5.2. Kiểm thử API 12**](#_Toc181292786)

[**1.5.3. Kiểm thử ứng dụng di động (Mobile Testing) 12**](#_Toc181292787)

[**1.5.4. Tích hợp DevOps và CI/CD 12**](#_Toc181292788)

[**1.5.5. Kiểm thử hồi quy (Regression Testing) 12**](#_Toc181292789)

[**1.5.6. Kiểm thử cross-browser và cross-platform 13**](#_Toc181292790)

[**CHƯƠNG 2. DÙNG KATALON ĐỂ KIỂM TRA CHỨC NĂNG ĐĂNG NHẬP 14**](#_Toc181292791)

[**2.1. DEMO 14**](#_Toc181292792)

[**2.1.1. Dữ liệu cần chuẩn bị 14**](#_Toc181292793)

[**2.1.2. Các bước tiến hành 15**](#_Toc181292794)

[**2.2. SO SÁNH VÀ KẾT LUẬN 34**](#_Toc181292795)

[**2.2.1. Điểm mạnh của Katalon so với các công cụ kiểm thử khác: 34**](#_Toc181292796)

[**2.2.2. Điểm yếu của Katalon so với các công cụ kiểm thử khác: 35**](#_Toc181292797)

[**2.2.3. Kết luận: 36**](#_Toc181292798)

**DANH MỤC HÌNH ẢNH**

[**Hình 1. Trang Download Katalon 3**](#_Toc181293031)

[**Hình 2. Giao diện Katalon Studio 4**](#_Toc181293032)

[**Hình 3. Thanh công cụ 4**](#_Toc181293033)

[**Hình 4. Text Explorer 5**](#_Toc181293034)

[**Hình 5. Manual Mode 6**](#_Toc181293035)

[**Hình 6. Script Mode 6**](#_Toc181293036)

[**Hình 7. Cửa sổ Log Viewer 7**](#_Toc181293037)

[**Hình 8. Cửa sổ Console 7**](#_Toc181293038)

[**Hình 9. Data File 8**](#_Toc181293039)

[**Hình 10. Dữ liệu file excel 15**](#_Toc181293040)

[**Hình 11. Tạo file Test Case 16**](#_Toc181293041)

[**Hình 12. Đặt tên file Test Case 17**](#_Toc181293042)

[**Hình 13. Record Web 17**](#_Toc181293043)

[**Hình 14. Dán đường dẫn của web sẽ test 18**](#_Toc181293044)

[**Hình 15. Giao diện web dùng để test 18**](#_Toc181293045)

[**Hình 16. Các thao tác đã được ghi lại 19**](#_Toc181293046)

[**Hình 17. Thêm trường hợp Delay và Verify 20**](#_Toc181293047)

[**Hình 18. Tạo các biến dùng để chứa dữ liệu test 20**](#_Toc181293048)

[**Hình 19. Gán biến Username vào bước kiểm thử 21**](#_Toc181293049)

[**Hình 20. Gán biến Password vào bước kiểm thử 22**](#_Toc181293050)

[**Hình 21. Gán biến Expected\_msg vào bước kiểm thử 22**](#_Toc181293051)

[**Hình 22. Tạo file chứa dữ liệu input 23**](#_Toc181293052)

[**Hình 23. Đặt tên file là DataLogin 23**](#_Toc181293053)

[**Hình 24. Tham chiếu tới file dữ liệu excel 24**](#_Toc181293054)

[**Hình 25. Kết quả của việc tham chiếu dữ liệu từ file excel 24**](#_Toc181293055)

[**Hình 26. Tạo file Test Suite dùng để quan sát kết quả kiểm thử 25**](#_Toc181293056)

[**Hình 27. Đặt tên file là Test 26**](#_Toc181293057)

[**Hình 28. Giao diện của file Test Suite 26**](#_Toc181293058)

[**Hình 29. Tham chiếu file Login vào file Test 27**](#_Toc181293059)

[**Hình 30. Show Data Binding 27**](#_Toc181293060)

[**Hình 31. Giao diện Show Data Binding 28**](#_Toc181293061)

[**Hình 32. Tham chiếu file DataLogin vào phần Test Data 28**](#_Toc181293062)

[**Hình 33. Kết quả sau khi tham chiếu file DataLogin vào phần Test Data 29**](#_Toc181293063)

[**Hình 34. Quét dữ liệu file DataLogin 29**](#_Toc181293064)

[**Hình 35. Run 30**](#_Toc181293065)

[**Hình 36. Kết quả hiển thị 30**](#_Toc181293066)

[**Hình 37. Xuất kết quả sang file PDF 31**](#_Toc181293067)

[**Hình 38. Kết quả chi tiết 32**](#_Toc181293068)

[**Hình 39. Kết quả chi tiết 32**](#_Toc181293069)

[**Hình 40. Kết quả chi tiết 33**](#_Toc181293070)

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. TỔNG QUÁT

### 1.1.1. Khái niệm

**Katalon Studio** là công cụ **tự động hóa kiểm thử phần mềm**, hỗ trợ kiểm thử ứng dụng web, di động và API. Nó có giao diện thân thiện, sử dụng ngôn ngữ Groovy, và tích hợp tốt với quy trình CI/CD, giúp cải thiện hiệu quả kiểm thử.

Katalon Studio là một công cụ giúp tự động hóa kiểm thử phần mềm. Nó cho phép bạn kiểm tra các ứng dụng web, di động và API một cách dễ dàng. Nói chung, Katalon Studio là một lựa chọn tốt cho những ai muốn kiểm thử phần mềm mà không cần quá nhiều kỹ năng chuyên môn.

### 1.1.2. Quá trình ra đời

Katalon Studio ra đời vào năm 2015, phát triển bởi KMS Technology, với mục tiêu cung cấp một giải pháp kiểm thử tự động toàn diện và dễ sử dụng. Trước khi Katalon xuất hiện, các công cụ kiểm thử tự động như Selenium và Appium đã phổ biến, nhưng chúng đòi hỏi người dùng phải có kỹ năng lập trình cao để có thể tạo và duy trì các bài kiểm thử. Điều này tạo ra rào cản lớn cho những đội ngũ kiểm thử không có nền tảng về lập trình.

### 1.1.3. Lý do ra đời

Đơn giản hóa quy trình kiểm thử tự động: Katalon Studio được phát triển để giúp cả người không có kinh nghiệm lập trình lẫn các lập trình viên kiểm thử dễ dàng tạo ra các kịch bản kiểm thử tự động. Giao diện người dùng thân thiện và hỗ trợ "record & playback" giúp người kiểm thử nhanh chóng tạo ra các bài kiểm thử mà không cần phải viết mã phức tạp.

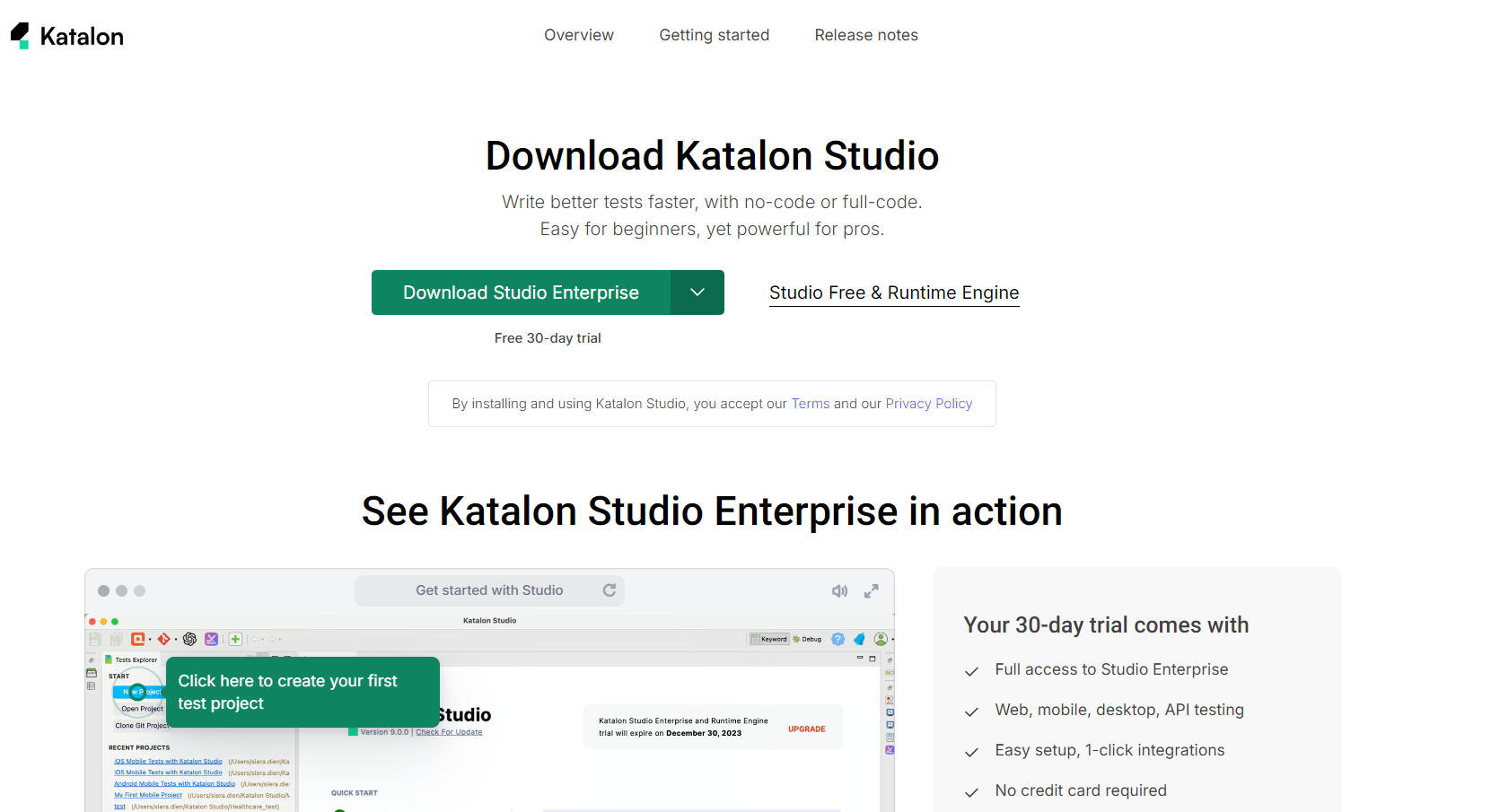
Tích hợp các công cụ kiểm thử phổ biến: Katalon Studio xây dựng trên nền tảng của Selenium và Appium, giúp người dùng có thể sử dụng sức mạnh của những công cụ này mà không cần phải trực tiếp làm việc với mã nguồn phức tạp. Nó cung cấp một môi trường trực quan, với các tính năng kéo thả và biểu đồ dễ sử dụng, giúp tự động hóa các bài kiểm thử web, ứng dụng di động và API.

Giảm chi phí và tăng tốc độ kiểm thử: Một trong những lý do Katalon được tạo ra là để giảm bớt chi phí phát triển và duy trì kiểm thử tự động so với việc sử dụng các công cụ kiểm thử mã nguồn mở đòi hỏi nhiều công sức và tài nguyên. Katalon Studio cũng hỗ trợ nhiều ngôn ngữ lập trình và nền tảng, từ đó tăng khả năng áp dụng rộng rãi cho các dự án khác nhau.

Tính năng toàn diện: Katalon Studio không chỉ hỗ trợ kiểm thử giao diện người dùng mà còn hỗ trợ kiểm thử API, dịch vụ web và ứng dụng di động. Điều này giúp các đội ngũ kiểm thử có thể kiểm thử đồng thời nhiều lớp của ứng dụng trong cùng một công cụ.

### 1.1.4. Cách tải

- [Link tải công cụ kiểm thử Katalon.](https://katalon.com/download)



Hình . Trang Download Katalon

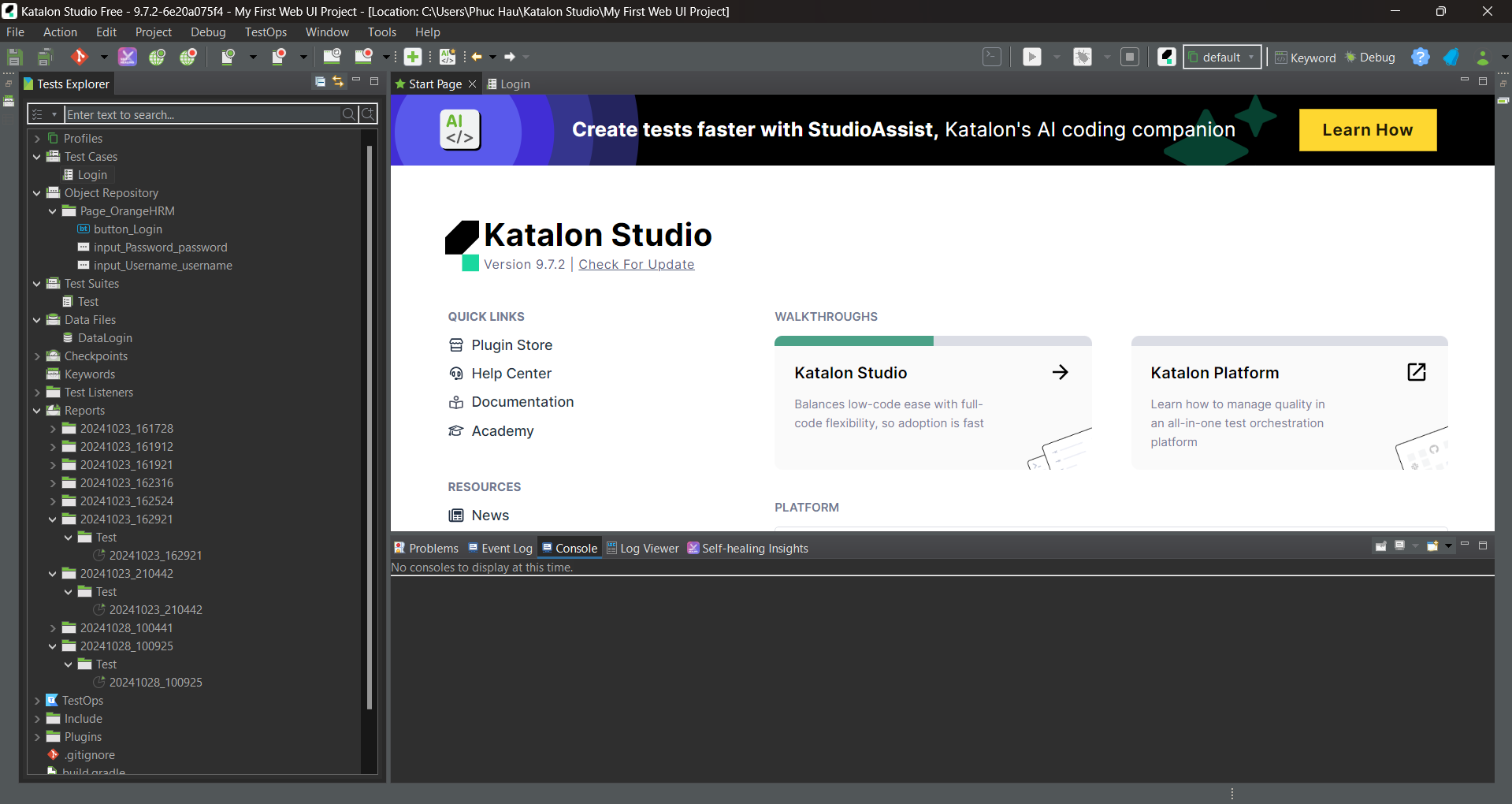
- Tìm phần “Katalon Studio Free” và chọn tải xuống.

- Sau khi ta nhấn tải xuống thì sẽ được file “KatalonSetup.exe”.

- Sau khi chạy file exe thì ta có thể bắt đầu sử dụng công cụ kiểm thử phần mềm Katalon.

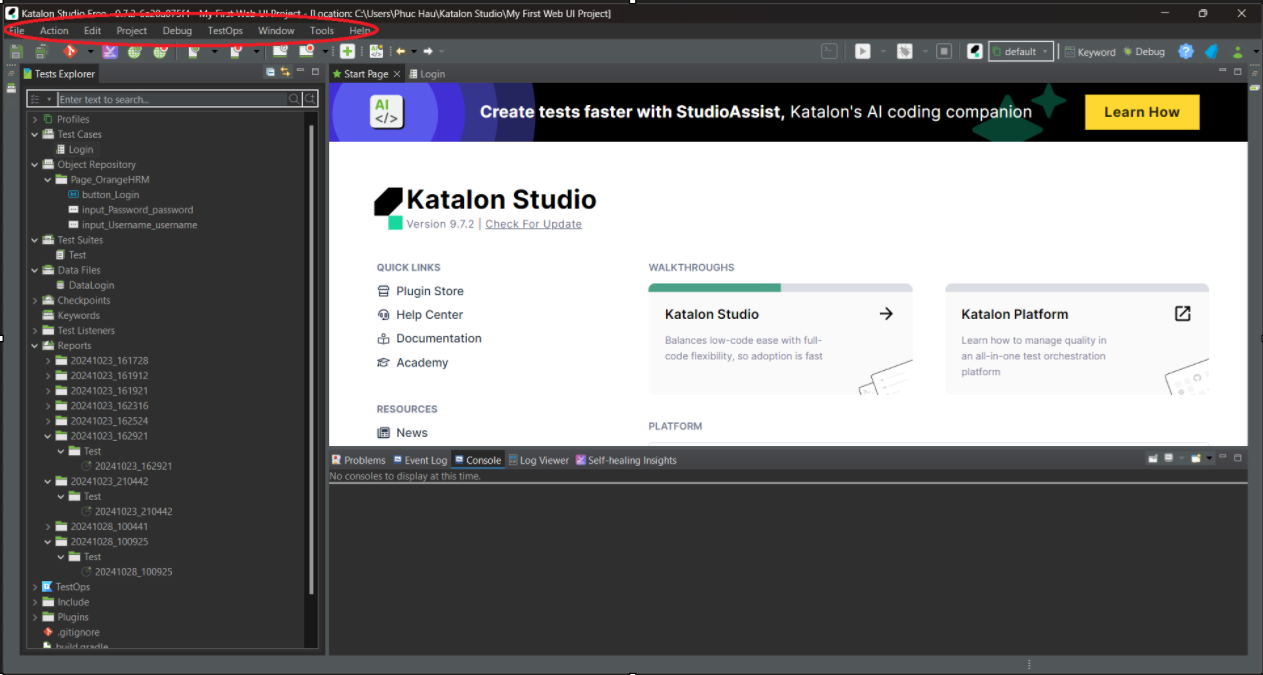
## 1.2. GIAO DIỆN

### 1.2.1. Tổng quan giao diện Katalon Studio



Hình 2. Giao diện Katalon Studio

**Thanh công cụ (Toolbar)**

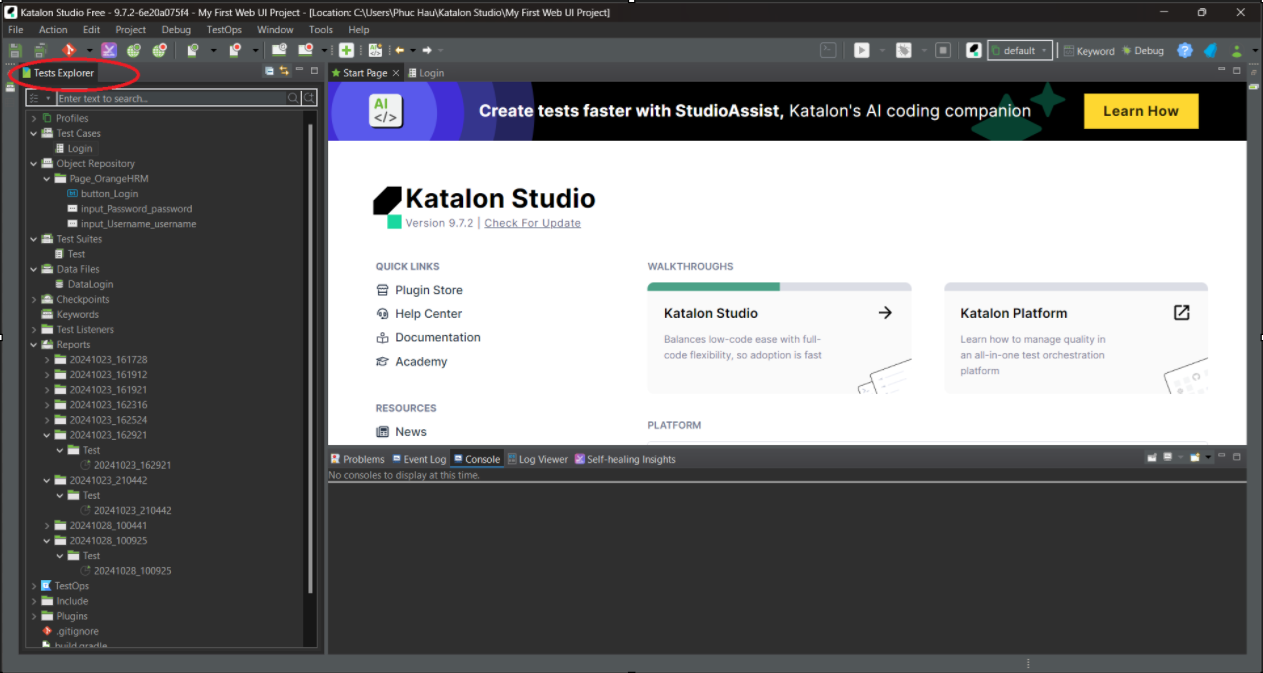


Hình . Thanh công cụ

* Nằm ở đầu giao diện, chứa các nút nhanh để tạo dự án mới, lưu dự án, chạy kiểm thử, dừng kiểm thử, và truy cập các công cụ khác.
* Cung cấp truy cập nhanh đến các tính năng phổ biến, giúp tiết kiệm thời gian khi thao tác.

**Test Explorer:** Nằm ở bên trái giao diện, Test Explorer hiển thị cấu trúc thư mục của dự án, giúp người dùng dễ dàng quản lý và tổ chức tất cả thành phần liên quan đến dự án kiểm thử, bao gồm:

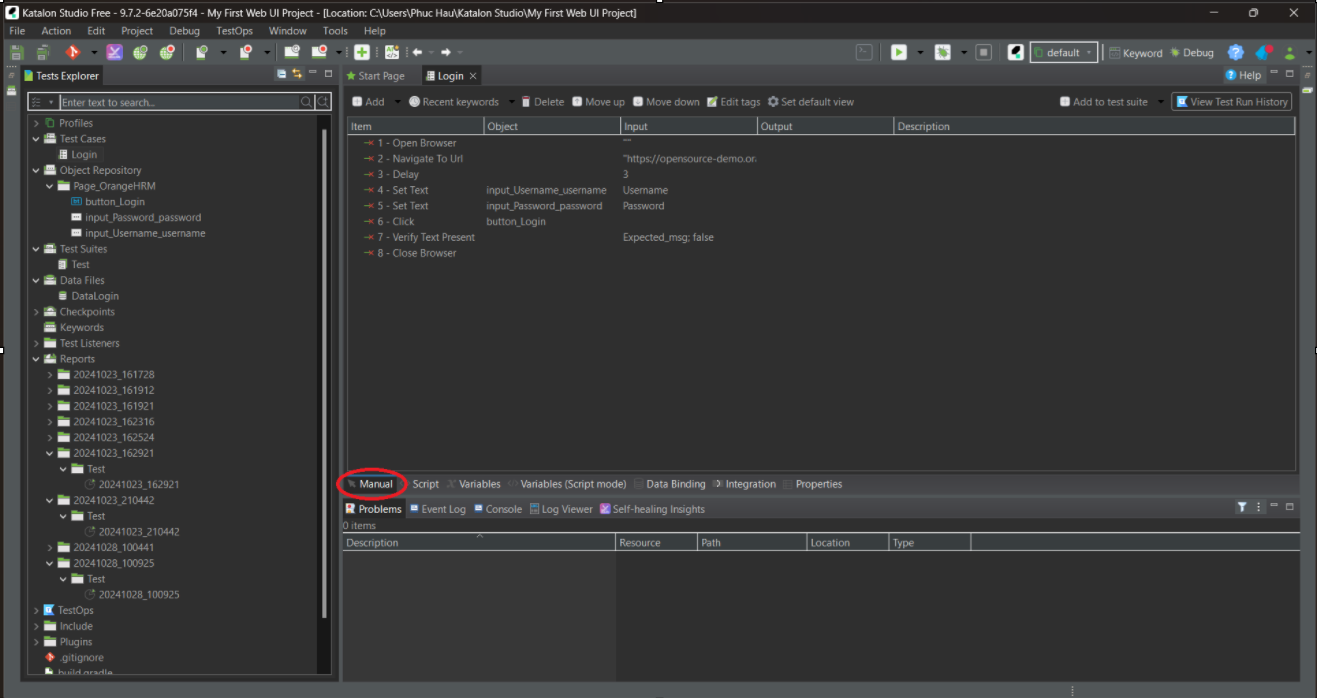
* **Test Cases**: Chứa các kịch bản kiểm thử, nơi người dùng tạo và quản lý các trường hợp kiểm thử.
* **Test Suites**: Gom nhóm các kịch bản kiểm thử vào một bộ kiểm thử để dễ dàng quản lý và thực thi.
* **Object Repository**: Lưu trữ các đối tượng giao diện (UI elements) mà kịch bản kiểm thử sẽ tương tác, cho phép sử dụng lại các đối tượng này trong nhiều kịch bản khác nhau.
* **Data Files**: Chứa dữ liệu kiểm thử (test data) để phục vụ kiểm thử dựa trên dữ liệu (data-driven testing), giúp kiểm thử một kịch bản với nhiều bộ dữ liệu khác nhau.

****

Hình . Text Explorer

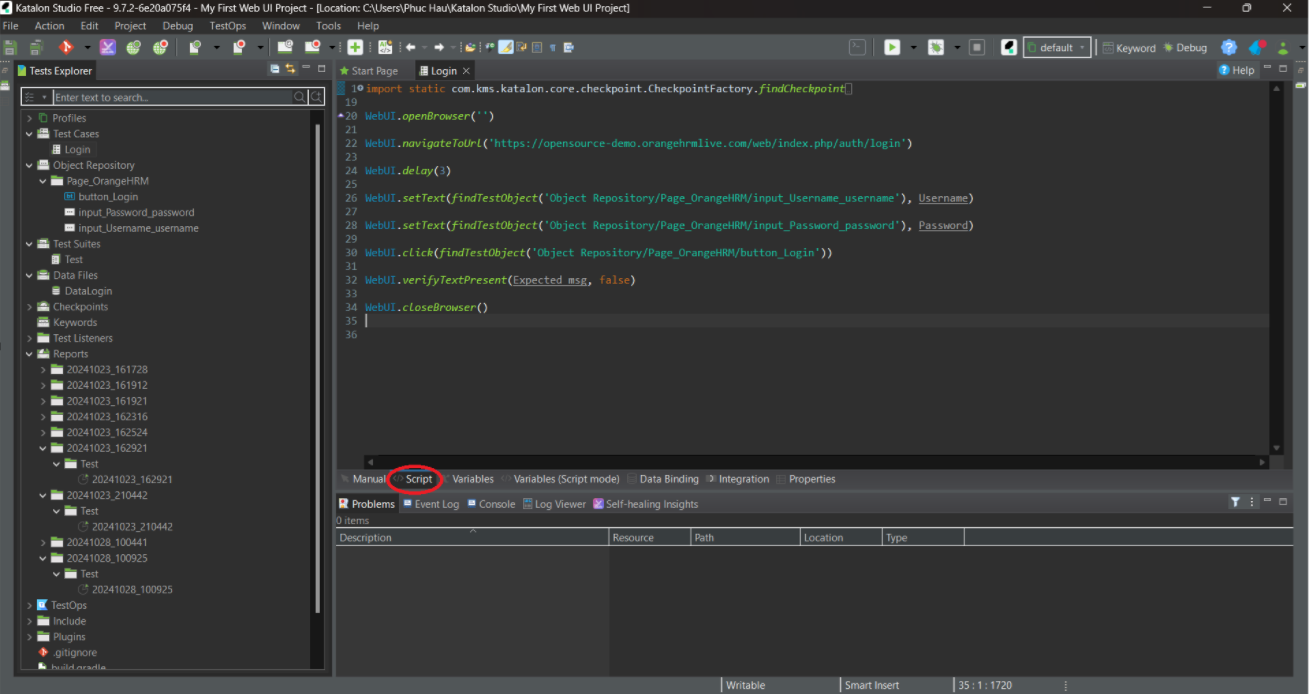
**Script và Manual Mode**: Có hai chế độ để viết kịch bản kiểm thử (lưu ý khi ta đã có file script trong mục Test Cases của cửa sổ Test Explorer thì **Script và Manual Mode mới được hiện):**

* **Manual Mode**: Cung cấp giao diện kéo-thả để xây dựng các bước kiểm thử mà không cần viết mã, phù hợp cho người dùng không chuyên về lập trình.



Hình . Manual Mode

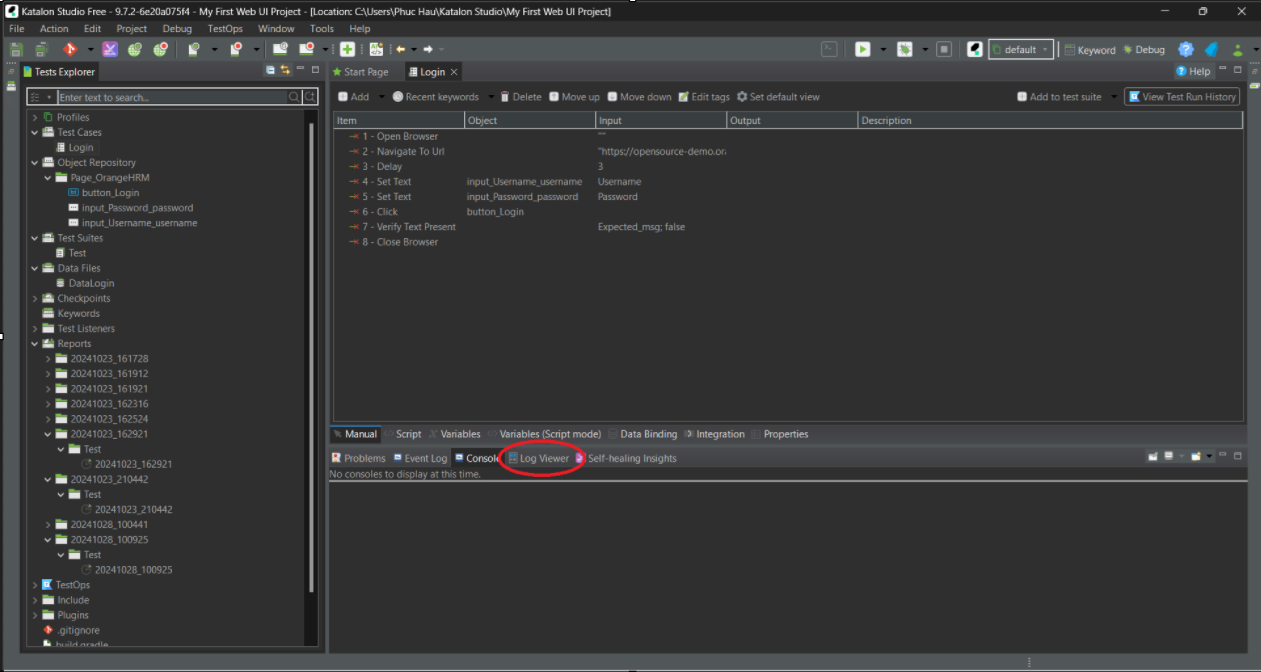
* **Script Mode**: Cho phép người dùng viết mã kiểm thử trực tiếp bằng Groovy hoặc Java, phù hợp cho người dùng có kiến thức lập trình.



Hình . ****Script** **Mode****

**Cửa sổ Log Viewer**:

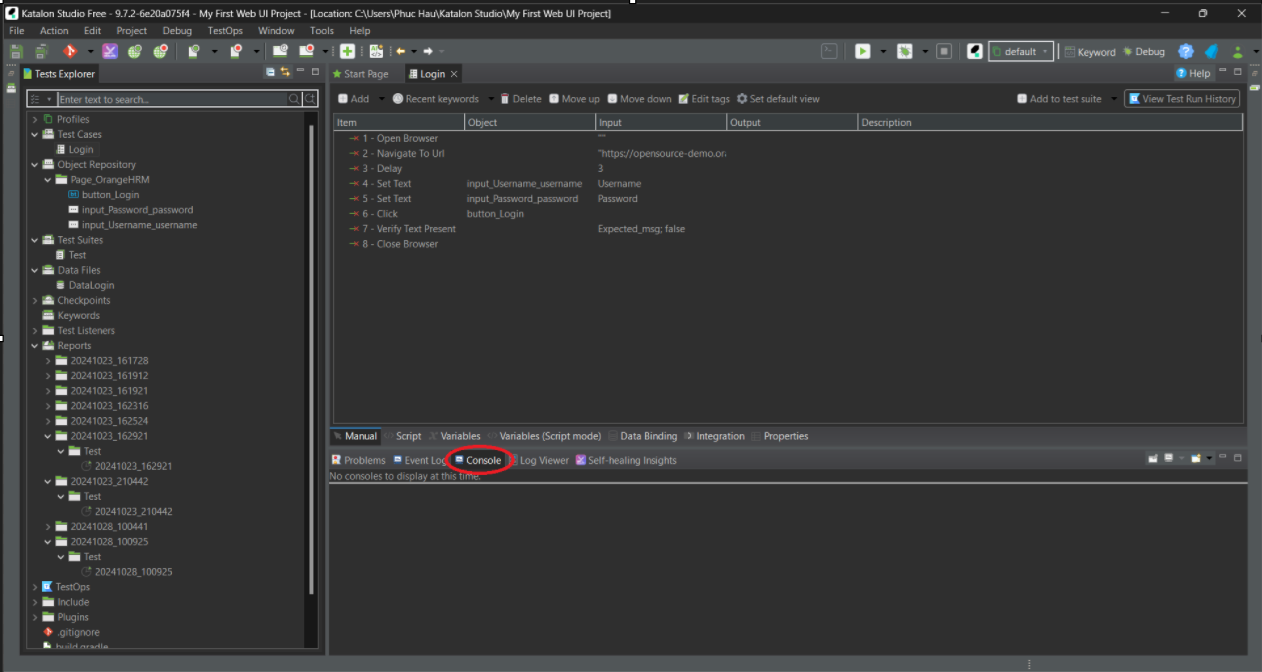
* Hiển thị chi tiết log về quá trình thực thi kịch bản kiểm thử, bao gồm thông báo thành công, cảnh báo, lỗi, và thông tin gỡ lỗi.
* Giúp người dùng xác định các lỗi xảy ra trong quá trình kiểm thử và kiểm tra các bước thực thi.



Hình . Cửa sổ Log ****Viewer****

**Cửa sổ Console**:

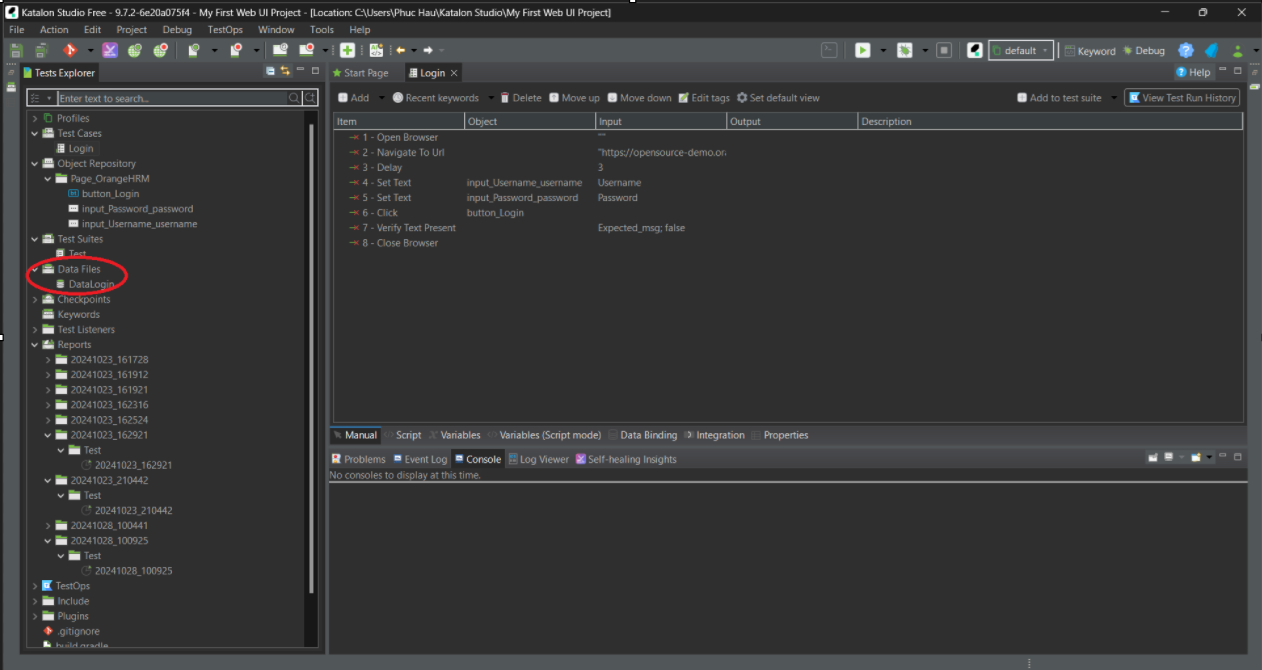
* Hiển thị các thông báo hệ thống và các log chi tiết khi chạy kiểm thử.
* Hữu ích cho việc gỡ lỗi và theo dõi chi tiết quá trình thực thi mã.

****

Hình . Cửa sổ Console

**Data File**:

* **Data Files** chứa các tệp dữ liệu kiểm thử từ nguồn bên ngoài như Excel, CSV hoặc Database, phục vụ cho việc kiểm thử dựa trên dữ liệu.



Hình . Data File

### 1.2.2. Các khái niệm liên quan

* **Test Case Editor**: Nơi bạn tạo và chỉnh sửa các kịch bản kiểm thử. Có thể ghi lại hành động hoặc viết mã bằng Groovy.
* **Object Spy:** Công cụ giúp bạn xác định và lưu trữ các đối tượng UI từ ứng dụng cần kiểm thử.
* **Data-Driven Testing:** Cho phép bạn kiểm thử với nhiều tập dữ liệu khác nhau từ các nguồn như Excel hoặc CSV.
* **Log Viewer:** Theo dõi log kiểm thử để kiểm tra các bước đã thực hiện và phát hiện lỗi.
* **Report Viewer:** Xem báo cáo chi tiết về kết quả kiểm thử, giúp bạn đánh giá hiệu suất và chất lượng sản phẩm.
* **Integration with CI/CD:** Katalon hỗ trợ tích hợp với các công cụ CI/CD như Jenkins, Bamboo, giúp tự động hóa quy trình kiểm thử.
* **API Testing:** Cung cấp công cụ để kiểm thử API, giúp bạn xác minh tính năng và hiệu suất của dịch vụ web.

## 1.3. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM

### 1.3.1. Ưu điểm

* **Miễn phí và mã nguồn mở**: Bạn có thể sử dụng Katalon Studio hoàn toàn miễn phí, thỏa sức khám phá mà không lo tốn kém với mã nguồn mở, thứ giúp bạn tiếp cận với muôn vàn kịch bản.
* **Hỗ trợ đa nền tảng**: Từ ứng dụng web, di động đến API, Katalon Studio có thể "chiều" bạn ở mọi lĩnh vực kiểm thử, cái gì khó đã có Katalon lo!
* **Tính năng ghi và phát lại**: Chỉ cần ghi lại các hành động, Katalon sẽ tự động biến chúng thành kịch bản kiểm thử.
* **Giao diện thân thiện**: Thiết kế hết sức thân thiện giúp người dùng dễ dàng làm quen mà không cần quá nhiều hướng dẫn, kể cả khi bạn không phải là dân chuyên!
* **Tích hợp dễ dàng**: Katalon có thể kết nối với nhiều công cụ CI/CD khác, giúp bạn gia tăng hiệu suất làm việc mà không gặp rắc rối.
* **Cộng đồng mạnh mẽ**: Có rất nhiều tài nguyên học tập và sự hỗ trợ từ cộng đồng người dùng, giúp bạn luôn có thể tìm thấy sự trợ giúp khi cần.

### 1.3.2. Nhược điểm

* **Hạn chế ngôn ngữ lập trình**: Katalon chủ yếu sử dụng Groovy(ngôn ngữ lập trình di động), nếu như bạn quen dung một ngôn ngữ khác thì chúc may mắn nhé.
* **Cần thời gian làm quen**: Một số tính năng nâng cao có thể cần một chút thời gian để tìm hiểu.
* **Tốc độ kiểm thử có thể chậm**: Nếu như là dự án lớn hoặc đơn giản là có nhiều luồng kịch bản, tốc độ thực hiện kiểm thử có thể không nhanh như mong đợi. Tuy nhiên, điều này cũng tùy thuộc vào cách bạn tối ưu hóa kịch bản, nếu bạn giỏi thì đây sẽ không phải một vấn đề quá lớn.
* **Hạn chế tính năng trong phiên bản miễn phí**: Mặc dù phiên bản miễn phí rất mạnh mẽ, nhưng một số tính năng cao cấp chỉ có trong phiên bản trả phí.

## 1.4. CÁCH SỬ DỤNG PHẦN MỀM

Dưới đây là các bước cơ bản hướng dẫn cách sử dụng phần mềm **Katalon**.

**1.** **Cài đặt và thiết lập ban đầu**

* **Tải và cài đặt Katalon**: Tải phần mềm từ trang chủ Katalon ([Katalon](https://katalon.com/download)) và cài đặt nó trên máy tính.
* **Đăng ký tài khoản**: Đăng nhập hoặc đăng ký tài khoản để kích hoạt bản quyền (bản miễn phí hoặc bản trả phí có thêm tính năng).

**2. Tạo dự án mới**

* Khi mở Katalon Studio, bạn sẽ tạo một dự án mới bằng cách vào File > New > Test Project.
* Chọn loại dự án (Web, API, Mobile, Desktop) và đặt tên cho dự án.

**3. Ghi lại các thao tác kiểm thử (Record & Playback)**

* Chọn **Record Web** (hoặc Mobile tùy vào nền tảng kiểm thử) để ghi lại các hành động trên trang web hoặc ứng dụng.
* Mọi thao tác (nhấp chuột, điền form, điều hướng trang) sẽ được ghi lại thành các bước trong kịch bản kiểm thử.
* Sau khi ghi, bạn có thể chỉnh sửa hoặc thêm các bước kiểm thử.

**4.** **Viết kịch bản kiểm thử bằng Groovy**

* Ngoài Record & Playback, bạn có thể viết kịch bản kiểm thử bằng ngôn ngữ Groovy (dựa trên Java).
* Katalon cung cấp các thư viện, hàm hỗ trợ sẵn giúp dễ dàng thao tác với các phần tử trên trang web, gọi API, và kiểm tra kết quả.

**5.** **Chạy kiểm thử**

* Sau khi tạo xong kịch bản kiểm thử (Test Case), nhấn nút **Run** và chọn môi trường (trình duyệt, thiết bị, API).
* Katalon sẽ tự động thực hiện các bước trong kịch bản và trả kết quả kiểm thử (thành công/thất bại) ngay trên giao diện.

**6.** **Kiểm thử API**

* Katalon hỗ trợ tạo các **API test cases** với các phương thức GET, POST, PUT, DELETE.
* Bạn có thể gọi các API endpoint, xác thực phản hồi và kiểm tra xem API hoạt động đúng không.

**7. Test Suite và Test Suite Collection**

* Test Suite: Tập hợp nhiều Test Case và chạy chúng theo trình tự.
* Test Suite Collection: Tập hợp nhiều Test Suite, giúp thực hiện kiểm thử với nhiều cấu hình và môi trường khác nhau.

**8.** **Báo cáo kết quả kiểm thử**

* Sau khi chạy, Katalon tự động tạo báo cáo chi tiết về kết quả của từng Test Case.
* Báo cáo bao gồm thông tin về từng bước kiểm thử, trạng thái (pass/fail), và thông tin lỗi (nếu có).

## 1.5. TÍNH ỨNG DỤNG CỦA KATALON STUDIO

**Katalon** có tính ứng dụng cao trong các lĩnh vực khác nhau nhờ khả năng tự động hóa các loại kiểm thử:

### 1.5.1. ****Kiểm thử giao diện web (Web UI Testing)****

* Katalon hỗ trợ kiểm thử trên các trình duyệt khác nhau như Chrome, Firefox, Edge, Safari.
* Dùng để kiểm thử các tương tác người dùng như điền form, nhấn nút, điều hướng trang, kiểm tra nội dung trang hiển thị.

### 1.5.2. ****Kiểm thử API****

* Với khả năng kiểm thử API mạnh mẽ, Katalon cho phép gọi các endpoint, gửi request, kiểm tra phản hồi và tích hợp với các dịch vụ RESTful và SOAP.
* Được ứng dụng trong các dự án phát triển microservices, dịch vụ web, và hệ thống phân tán.

### 1.5.3. ****Kiểm thử ứng dụng di động (Mobile Testing)****

* Katalon hỗ trợ kiểm thử các ứng dụng mobile trên Android và iOS, bao gồm cả ứng dụng gốc (native) và ứng dụng lai (hybrid).
* Công cụ này có thể ghi lại các thao tác trên thiết bị di động, hoặc giả lập thông qua môi trường giả lập.

### 1.5.4. ****Tích hợp DevOps và CI/CD****

* Trong quá trình phát triển phần mềm, Katalon giúp tự động hóa các bài kiểm thử trong pipeline CI/CD, đảm bảo ứng dụng luôn được kiểm tra mỗi khi có thay đổi về mã nguồn.
* Công cụ này có thể tích hợp với Jenkins, Azure DevOps, GitLab CI/CD và nhiều hệ thống khác để kiểm thử tự động.

### 1.5.5. ****Kiểm thử hồi quy (Regression Testing)****

* Katalon có thể lưu trữ và chạy lại các kịch bản kiểm thử đã có, giúp kiểm tra xem có tính năng nào bị ảnh hưởng sau khi cập nhật hoặc thêm tính năng mới.
* Điều này rất hữu ích trong các dự án phát triển dài hạn, cần duy trì chất lượng và ổn định của sản phẩm.

### 1.5.6. ****Kiểm thử cross-browser và cross-platform****

* Katalon giúp kiểm thử ứng dụng trên nhiều nền tảng và trình duyệt khác nhau, đảm bảo rằng ứng dụng hoạt động mượt mà và không có lỗi trên các môi trường khác nhau.
* Điều này rất quan trọng đối với các ứng dụng web và mobile mà người dùng sử dụng trên nhiều loại thiết bị và trình duyệt khác nhau.

# CHƯƠNG 2. DÙNG KATALON ĐỂ KIỂM TRA CHỨC NĂNG ĐĂNG NHẬP

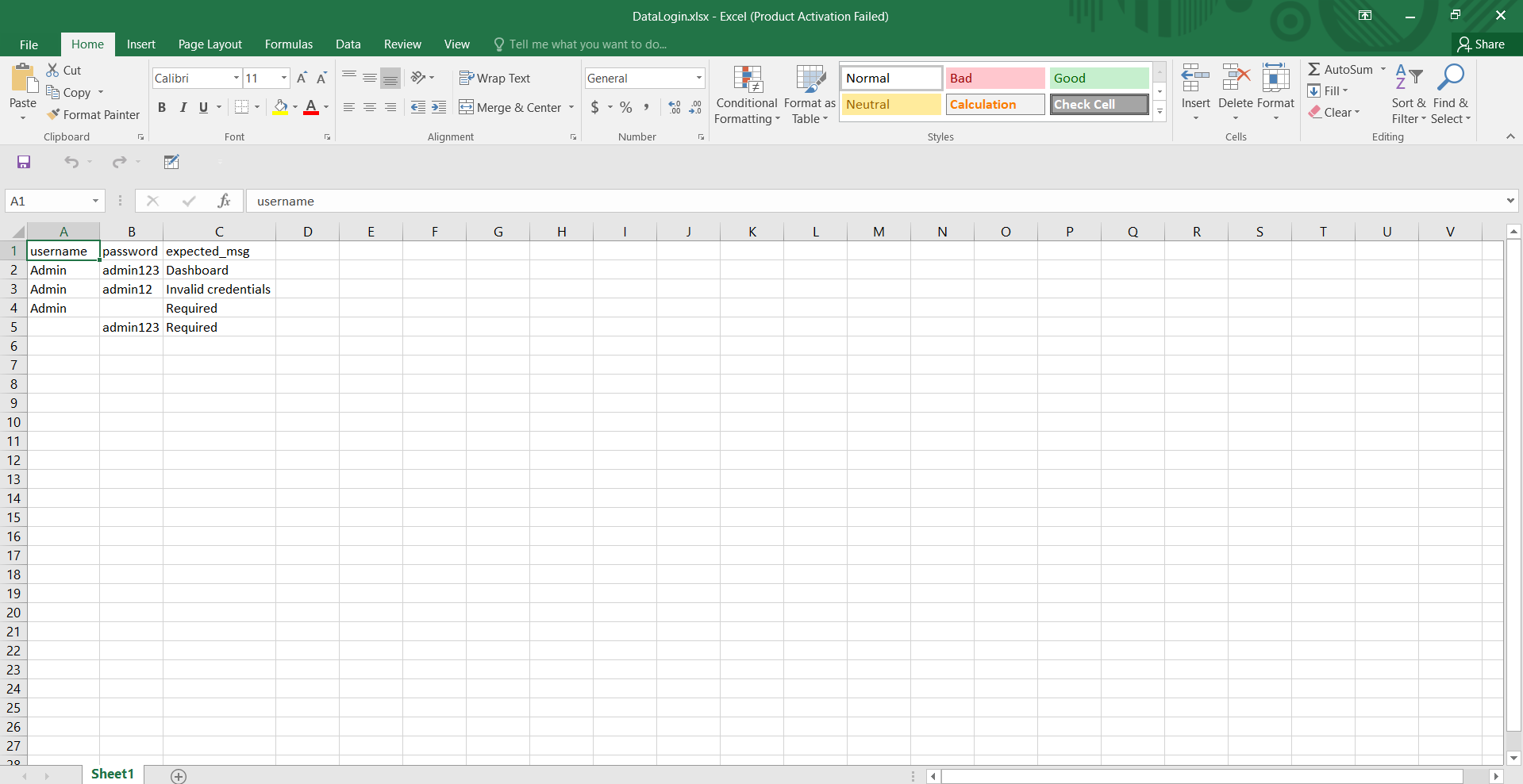
## 2.1. DEMO

### 2.1.1. Dữ liệu cần chuẩn bị

Trước khi bắt đầu kiểm thử thì ta cần chuẩn bị 1 file excel với dữ liệu như hình bên dưới.

Trước khi bắt đầu kiểm thử thì ta cần chuẩn bị 1 file chứa dữ liệu. Ta sẽ tạo 1 file excel có tên là DataLogin có chứa dữ liệu như hình bên dưới, với các trường dữ liệu là:

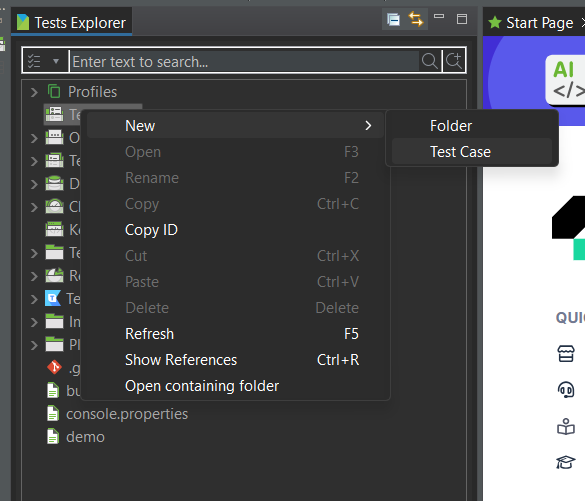
* Username: dùng để lưu trữ dữ liệu đầu vào của username.
* Password: dùng để lưu trữ dữ liệu đầu vào của password.
* Expected\_msg: lưu trữ đoạn text cần quét trên màn hình khi thực hiện đăng nhập (ví dụ như khi ta đăng nhập thất bại vì thiếu thông tin thì màn hình sẽ hiển thị chữ “Required”, còn nếu đăng nhập sai Username hoặc password thì sẽ là “Invalid credentials”…



Hình . Dữ liệu file excel

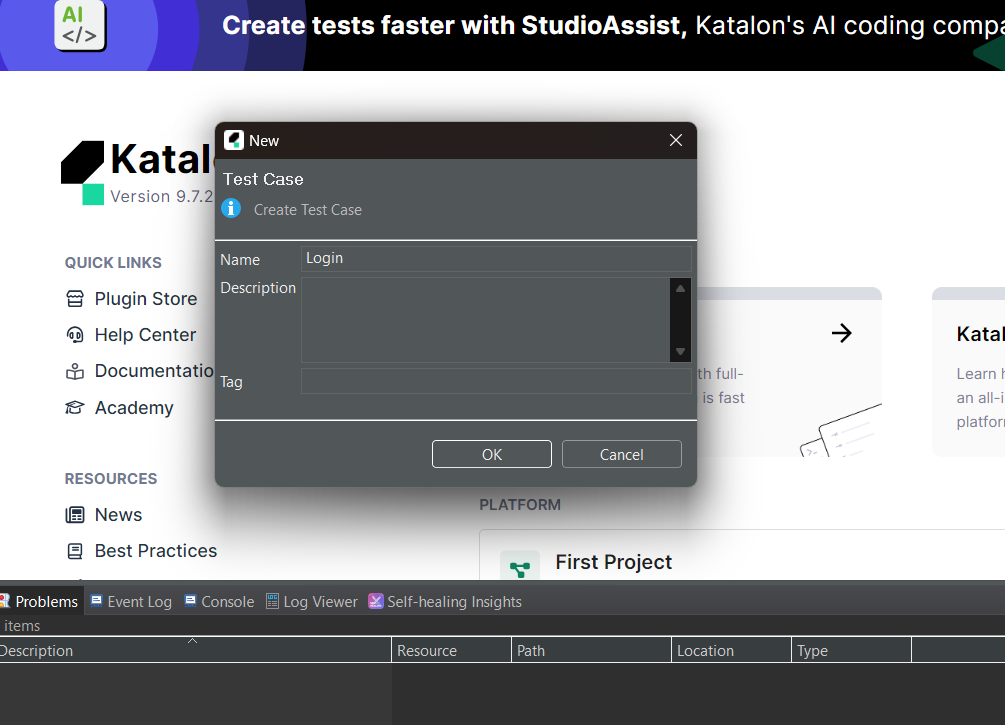
### 2.1.2. Các bước tiến hành

Đầu tiên ta tiến hành tạo file test case bằng cách nhấn chuột phải vào phần Test Cases trong Tests Explorer, chọn New -> Test Case.



Hình 11. Tạo file Test Case

Ta đặt tên file test là Login.



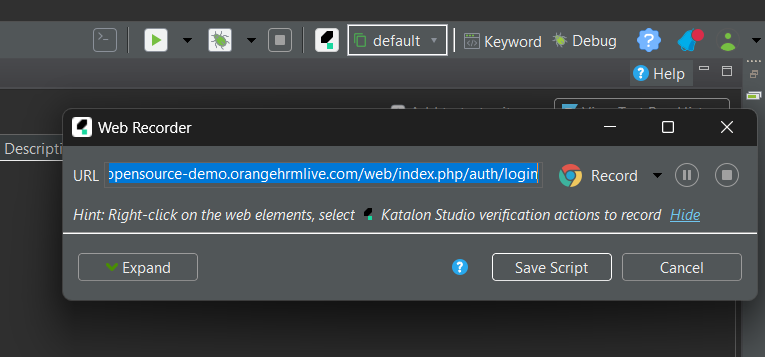
Hình 12. Đặt tên file Test Case

Sau đó ta tiến hành quá trình test bằng cách nhấn vào biểu tượng hình tròn màu xanh có nút đỏ trên thanh công cụ để Record Web.



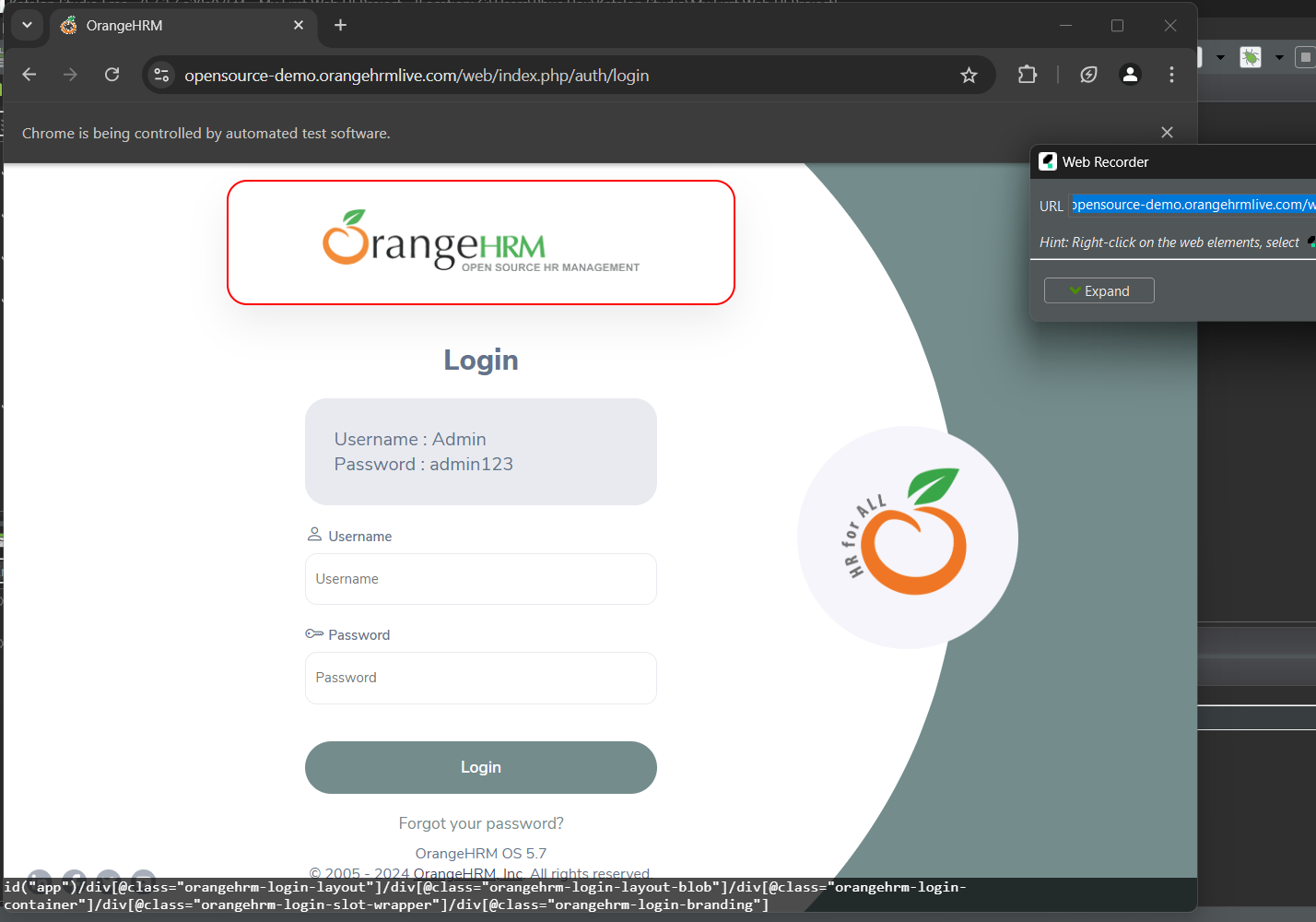
Hình 13. Record Web

Trong bài này ta sẽ dùng trang web open source của OrangeHRM để test, sau khi nhấn vào Record Web thì ta sẽ dán đường link [Trang OrangeHRM để test](https://opensource-demo.orangehrmlive.com/web/index.php/auth/login) vào phần URL.



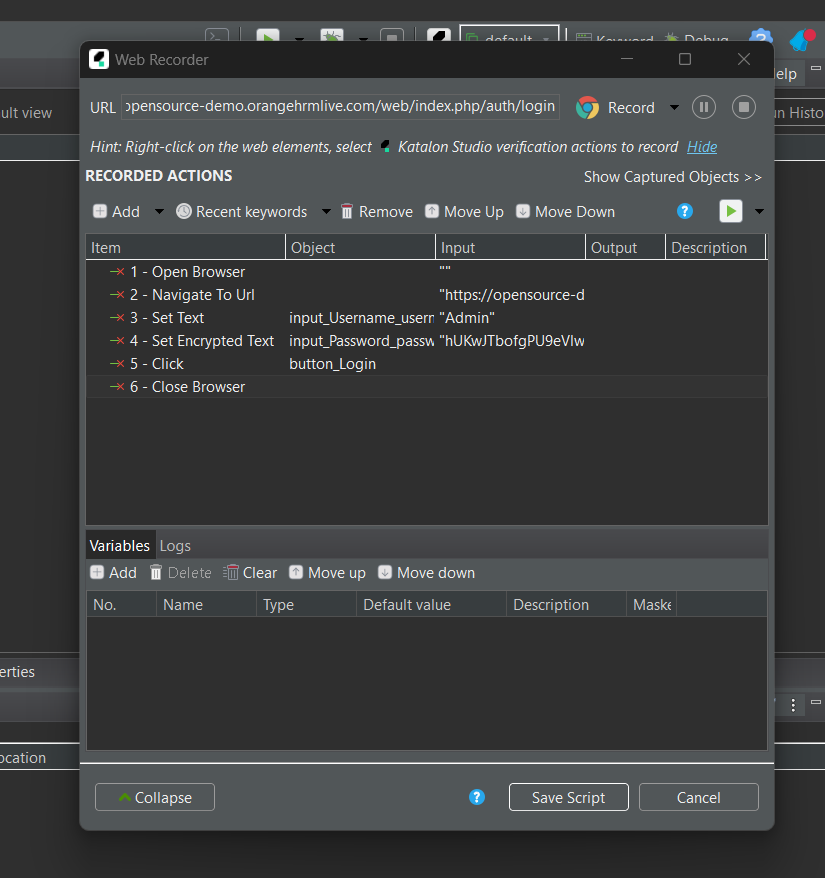
Hình 14. Dán đường dẫn của web sẽ test

Sau đó ta nhấn vào nút Record bên cạnh biểu tượng Chrome để bắt đầu, và một giao diện như hình bên dưới sẽ hiện lên để ta có thể ghi lại các thao tác trong quá trình test.



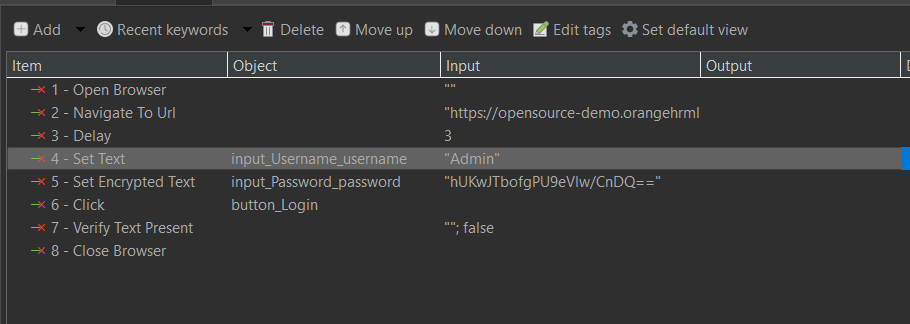
Hình 15. Giao diện web dùng để test

Ta sẽ nhập Username và Password đã được gợi ý để đăng nhập, sau khi ta đăng nhập xong và nhấn nút dừng lại thì một bảng mô tả các hành động đã được ghi lại sẽ hiển thị như hình bên dưới.



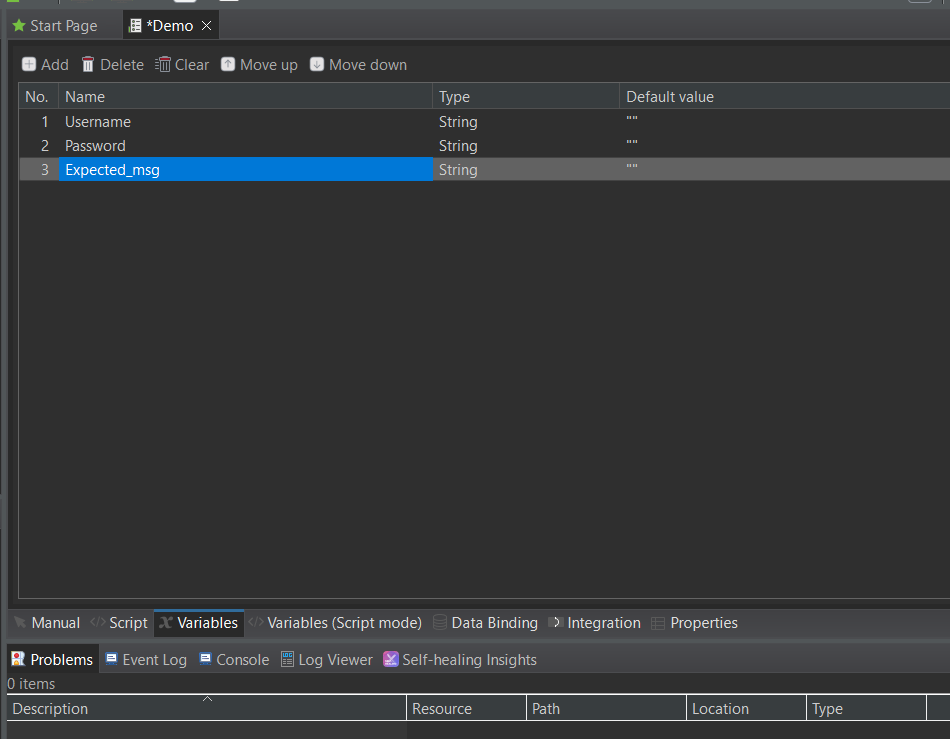
Hình 16. Các thao tác đã được ghi lại

Ta sẽ lưu các thao tác này lại để thực hiện điều chỉnh thêm. Ta sẽ tiến hành thêm các trường hợp như như là mạng yếu thì sẽ delay, và trường hợp verify Username và Password. Ta sẽ thêm 2 trường hợp này bằng nút Add và kết quả sẽ như sau.



Hình 17. Thêm trường hợp Delay và Verify

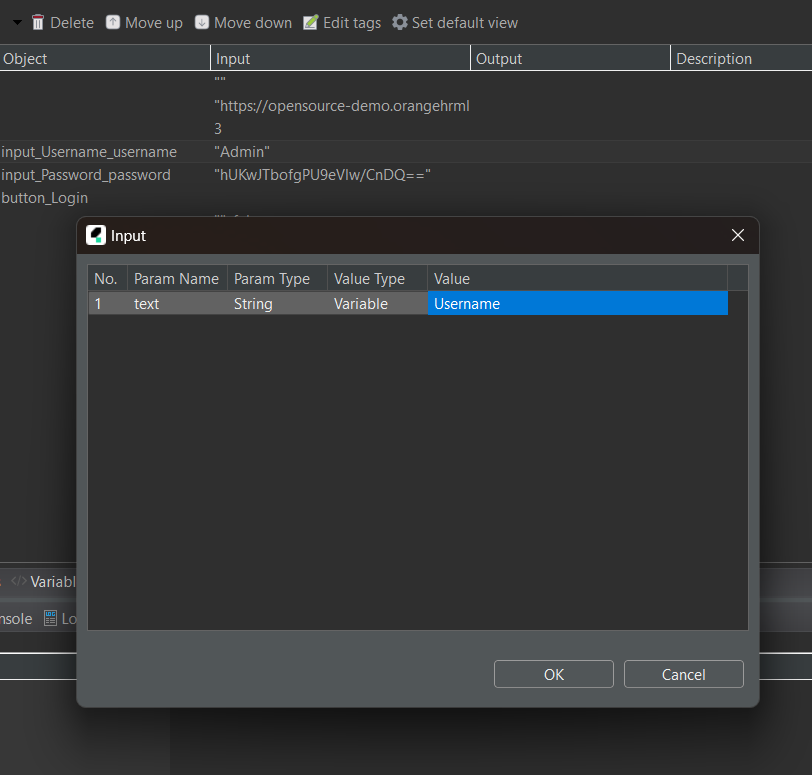
Tiếp theo ta sẽ vào tab Variables để tạo 3 biến lần lượt là Username, Password và Expected\_msg.



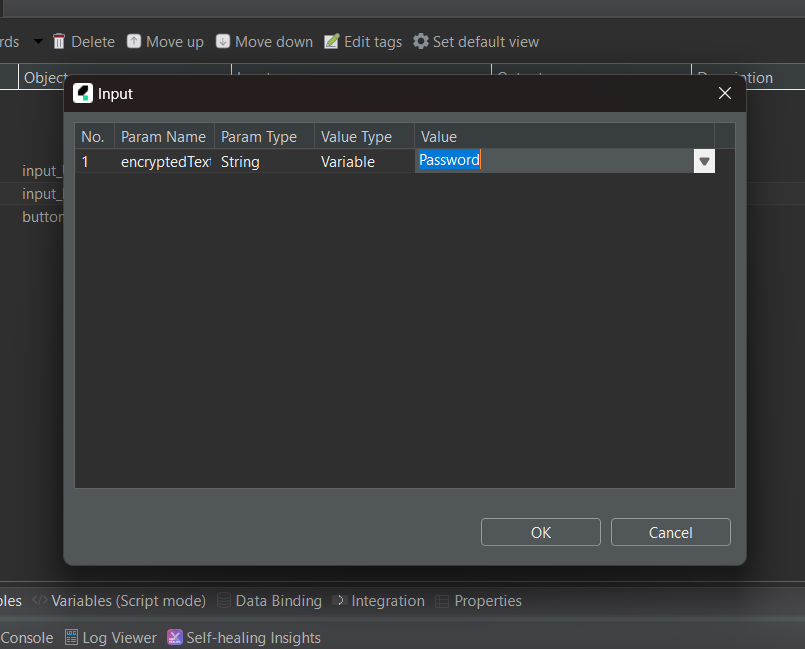
Hình 18. Tạo các biến dùng để chứa dữ liệu test

Ta sẽ gán các biến vừa tạo cho các bước trong quá trình kiểm thử (Username=>Set Text ở bước 4, Password=> Set Text ở bước 5, Expected\_msg => Verify Text Present).

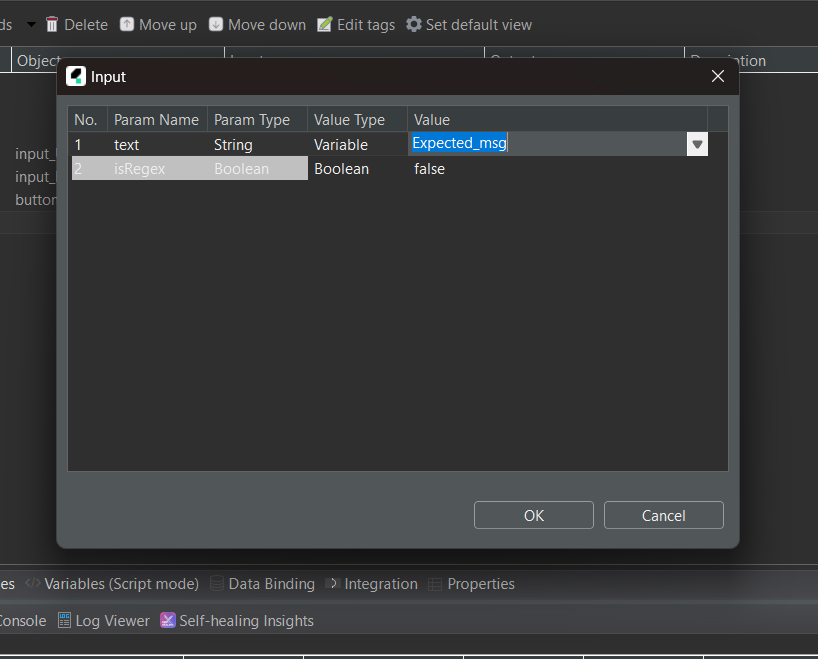
Ta click vào mục input của bước để gán giá trị, ta sẽ điều chỉnh Value Type và Value của chúng.



Hình 19. Gán biến Username vào bước kiểm thử

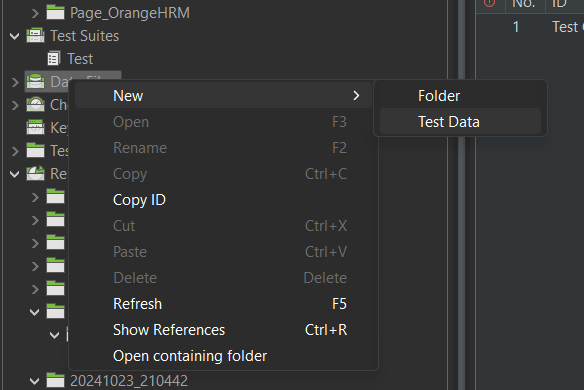


Hình 20. Gán biến Password vào bước kiểm thử



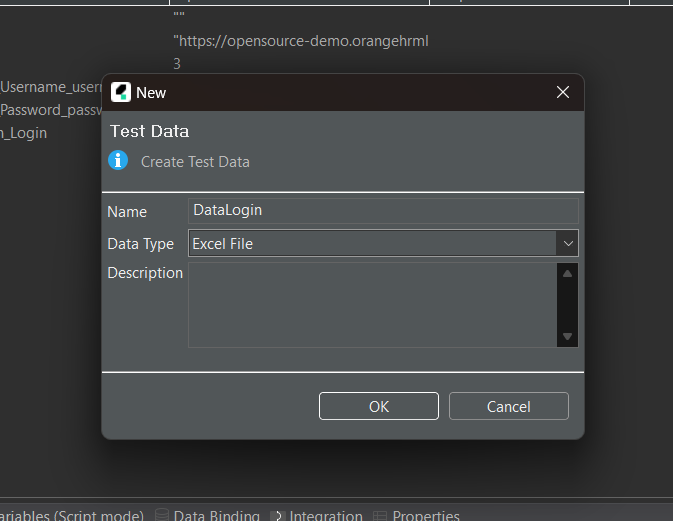
Hình 21. Gán biến Expected\_msg vào bước kiểm thử

Sau khi điều chỉnh những bước dùng để chạy test case xong thì ta sẽ tiến hành tạo file input, ta sẽ dùng file excel để làm danh sách input. Ta nhấn chuột phải vào mục Data Files trong Tests Explorer -> chọn New -> chọn Test Data.



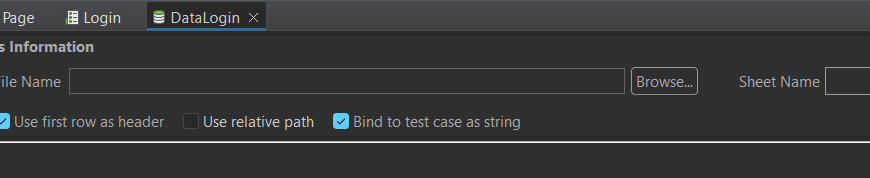
Hình 22. Tạo file chứa dữ liệu input

Ta đặt tên file là DataLogin



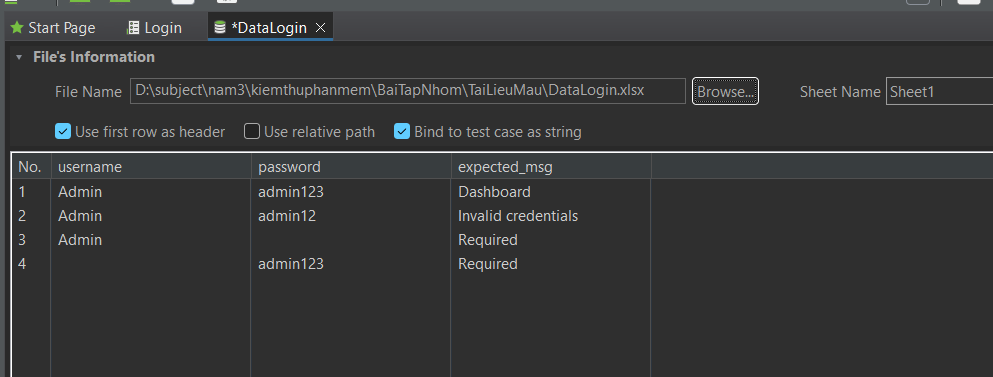
Hình 23. Đặt tên file là DataLogin

Sau đó ta nhấn vào mục Browse của file DataLogin và chọn file Excel đã được tạo trước ở mục 2.1.1. Dữ liệu cần chuẩn bị để tham chiếu dữ liệu cho việc kiểm thử vào file DataLogin này.



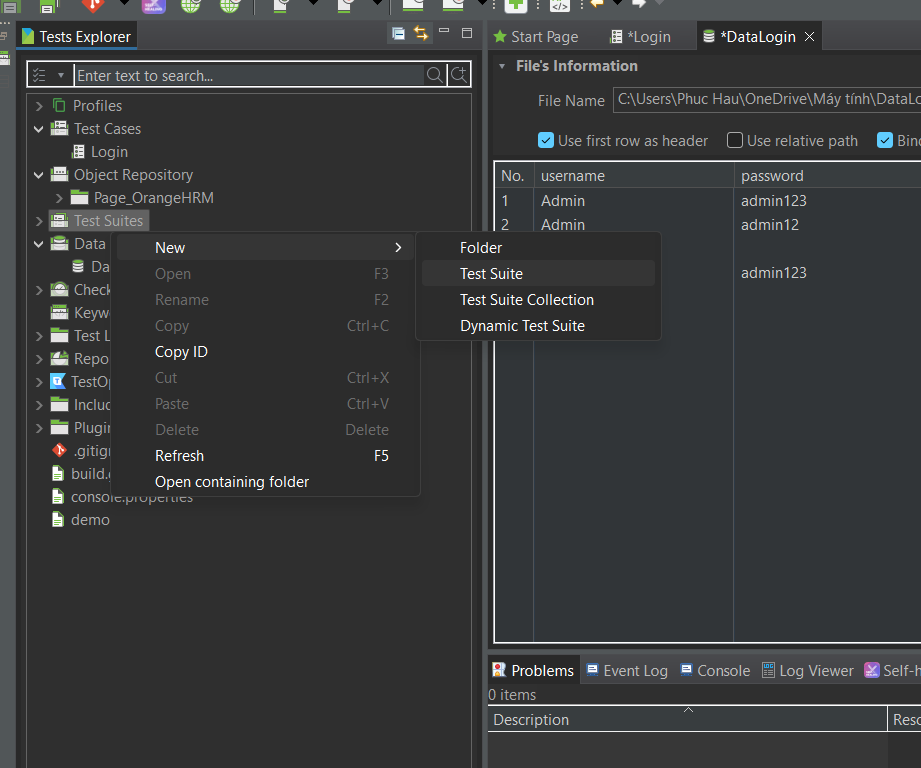
Hình 24. Tham chiếu tới file dữ liệu excel

Dữ liệu của file excel sẽ được tham chiếu vào file DataLogin như hình bên dưới.



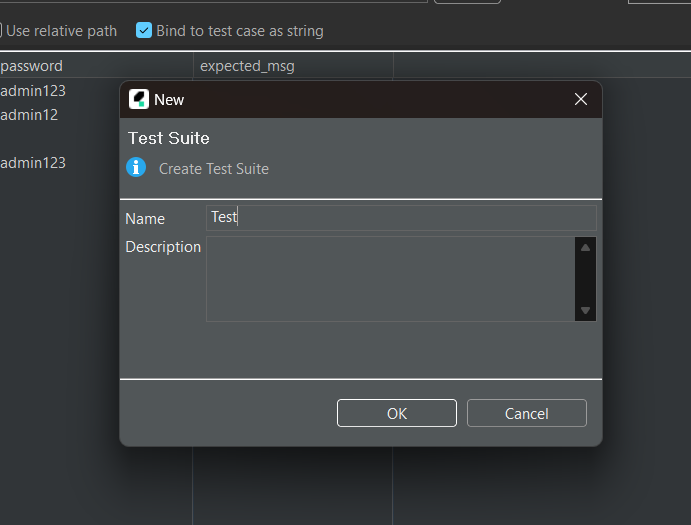
Hình 25. Kết quả của việc tham chiếu dữ liệu từ file excel

Sau đó ta sẽ tạo 1 file trong mục Test Suites dùng để quan sát kết quả kiểm thử, trong mục Test Suites ta nhấn chuột phải -> chọn New -> chọn Test Suite, đặt tên file là Test.



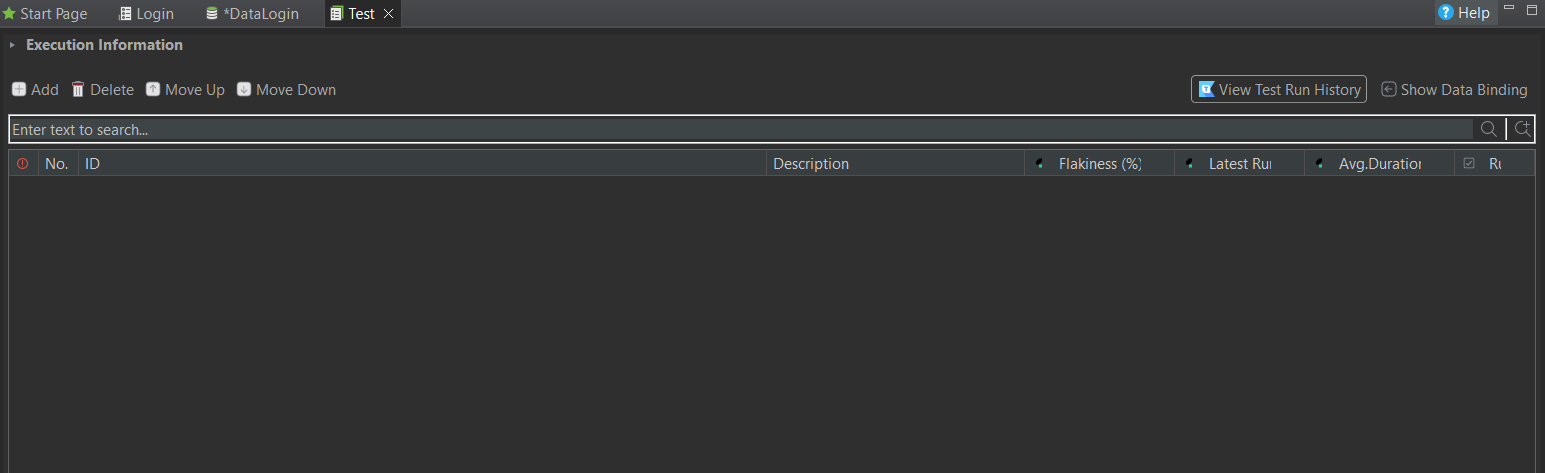
Hình 26. Tạo file Test Suite dùng để quan sát kết quả kiểm thử

Đặt tên file là Test.



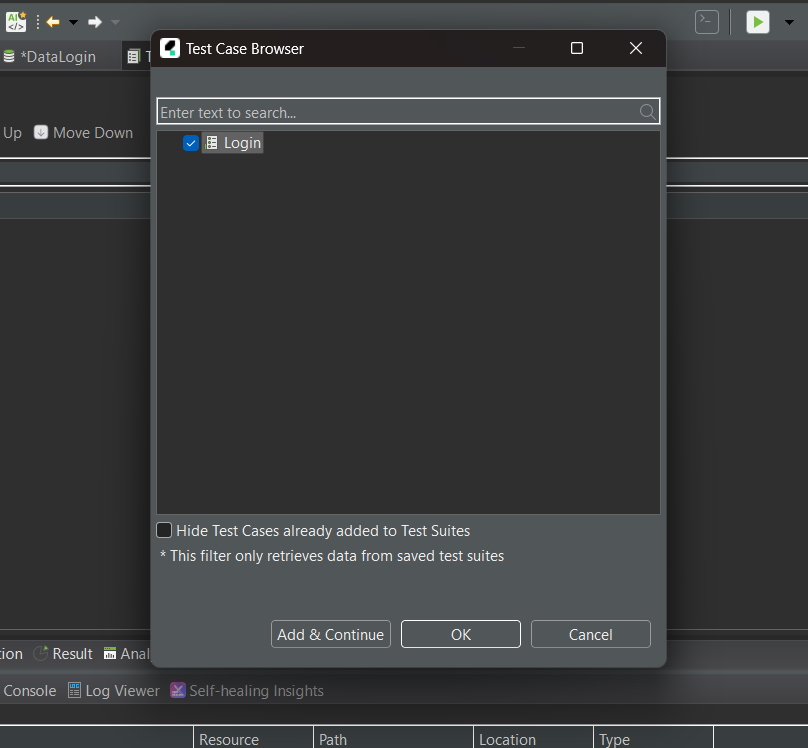
Hình 27. Đặt tên file là Test

Giao diện của test suite sau khi được tạo ra.



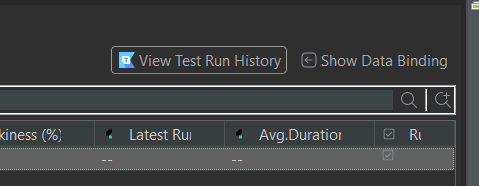
Hình 28. Giao diện của file Test Suite

Ta sẽ add file Login trong mục Test Cases vào file Test suite này bằng nút Add.



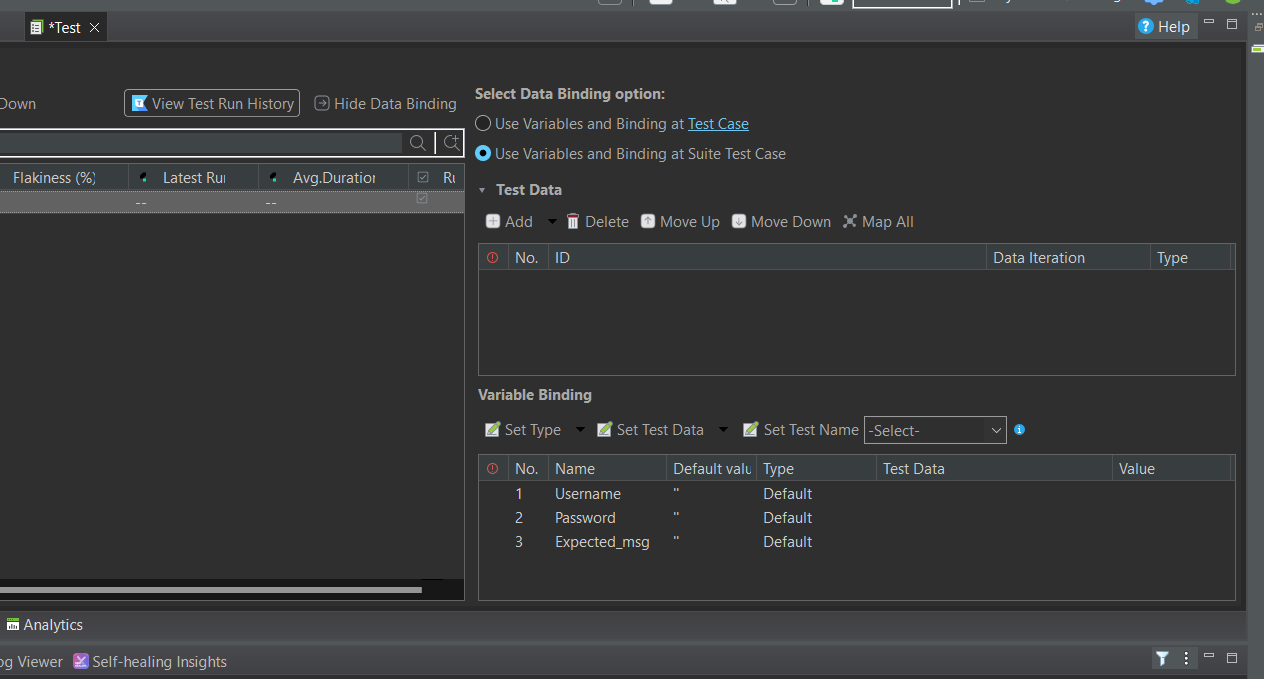
Hình 29. Tham chiếu file Login vào file Test

Sau đó ta nhấn vào mục Show Data Binding để hiện ra các mục data cần thiết cho quá trình test.



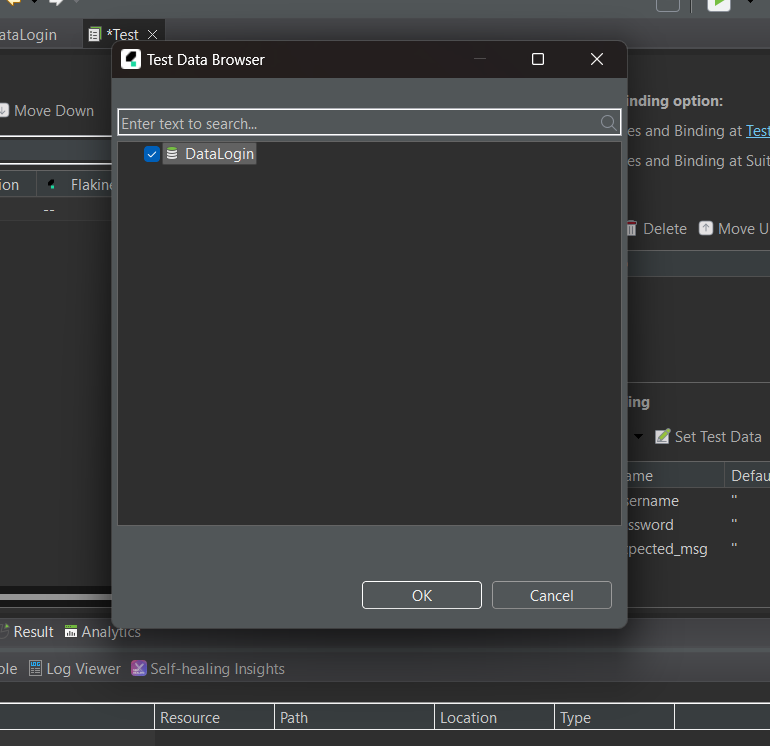
Hình 30. Show Data Binding

Một giao diện như hình dưới sẽ được hiện ra.



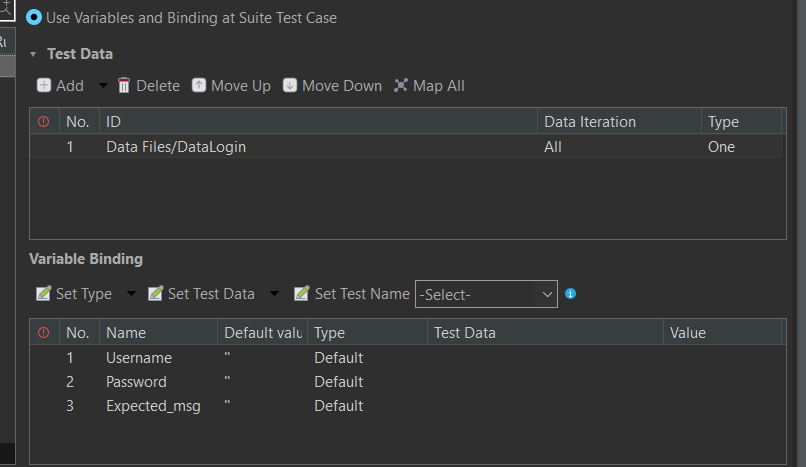
Hình 31. Giao diện Show Data Binding

Sau đó ta thêm dữ liệu vào bằng nút Add trong phần Test Data, ta sẽ thêm file excel DataLogin trong mục Data Files.



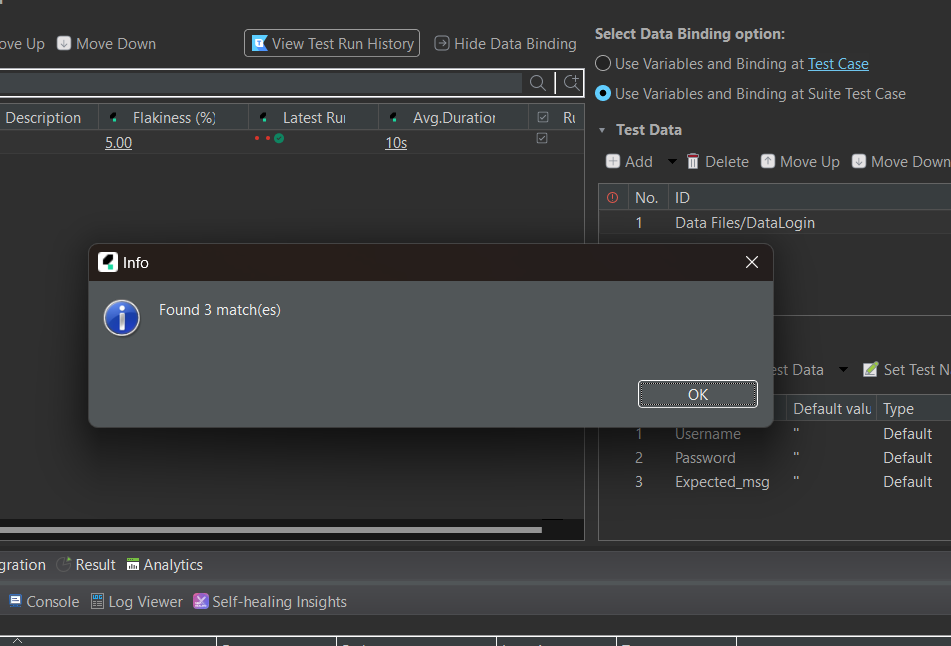
Hình 32. Tham chiếu file DataLogin vào phần Test Data

Kết quả sẽ như sau.



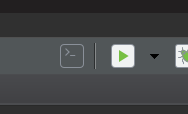
Hình 33. Kết quả sau khi tham chiếu file DataLogin vào phần Test Data

Ta sẽ nhấn vào mục Map All để quét dữ liệu của file DataLogin, một thông báo sẽ hiện ra khi quét dữ liệu thành công.

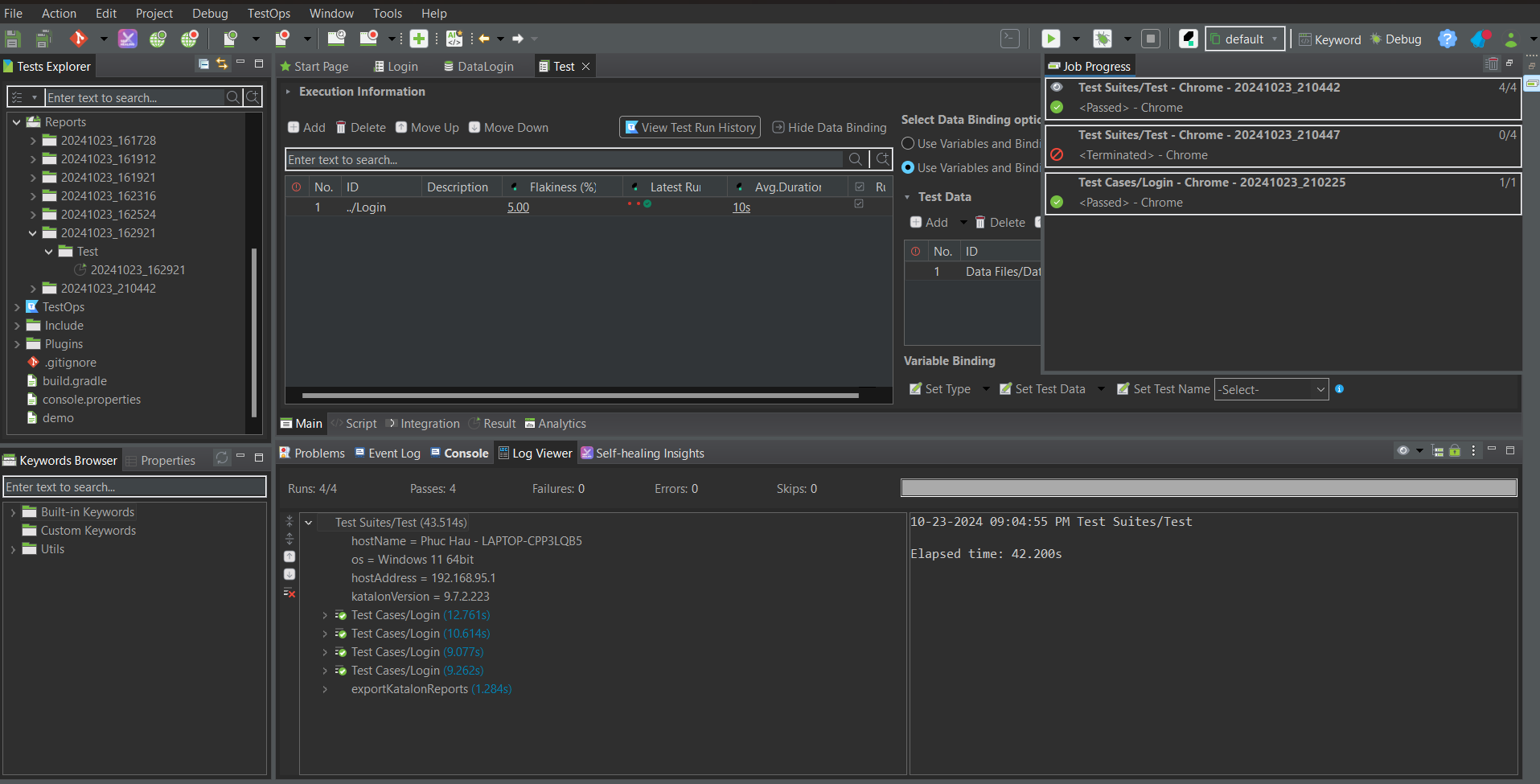


Hình 34. Quét dữ liệu file DataLogin

Sau đó ta quay lại file Test của mục Test Suite để chạy, ta nhấn vào biểu tượng Run để chạy.

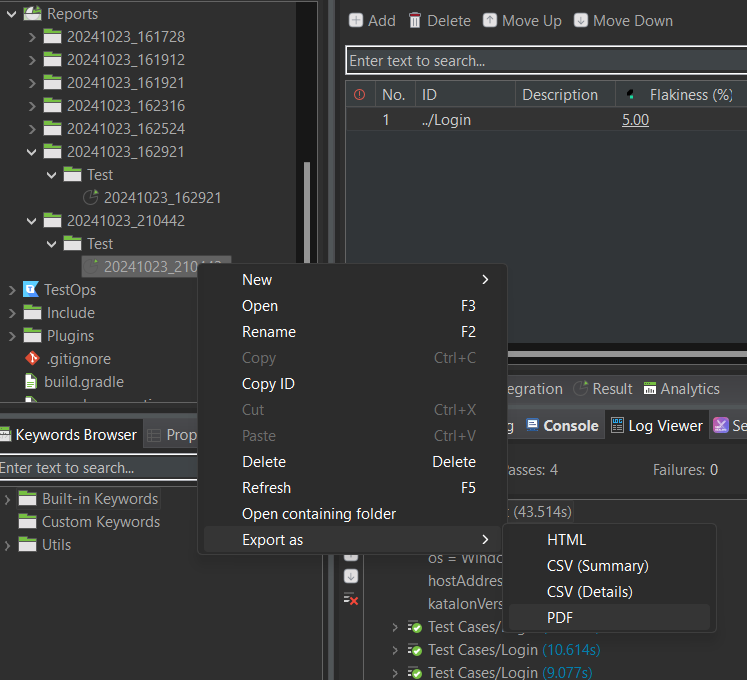


Hình 35. Run

Và kết quả sẽ được hiển thị trong tab Log Viewer và được lưu vào mục Reports. 

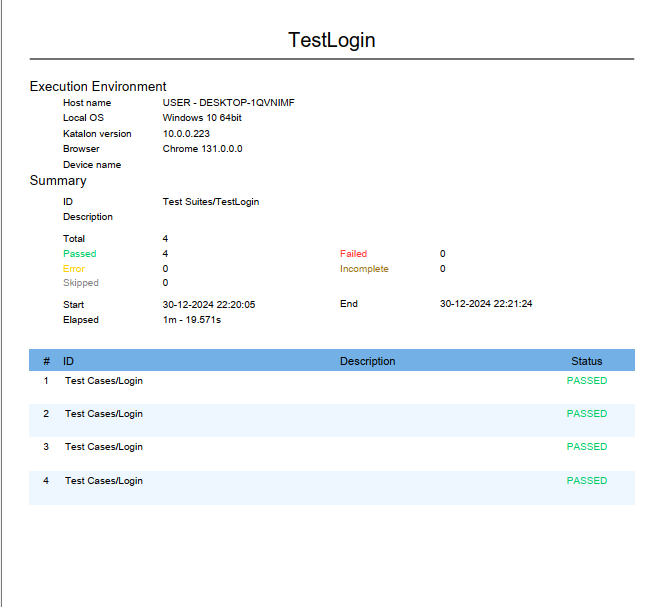
Hình 36. Kết quả hiển thị

Ta có thể xuất file PDF từ trong mục Reports để có thể xem chi tiết dữ liệu, ta sẽ chọn theo đường dẫn sau đây.

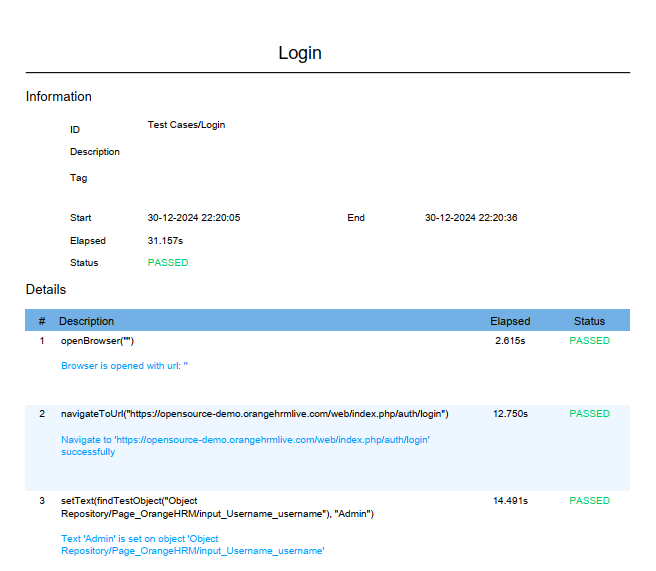


Hình 37. Xuất kết quả sang file PDF

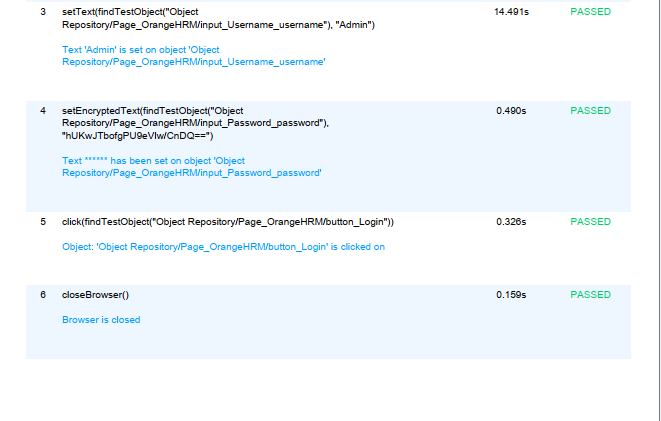
Kết quả chi tiết.



Hình 38. Kết quả chi tiết



Hình . Kết quả chi tiết



Hình . Kết quả chi tiết

**Chú ý**: ta phải điều chỉnh Item ở bước 5 từ Set Encrypted Text thành Set Text, nếu không điều chỉnh như này thì khi input ta đưa vào test sẽ bị mã hóa và dẫn đến kết quả sai. Và sau mỗi thao tác với mỗi file thì ta cần phải lưu file lại để việc kiểm tra diễn ra đúng và không xảy ra sự cố ngoài ý muốn.

## 2.2. SO SÁNH VÀ KẾT LUẬN

So sánh **Katalon** với các công cụ kiểm thử khác như Selenium, Appium, TestComplete, và Ranorex, dưới đây là điểm mạnh và điểm yếu của Katalon:

### 2.2.1. Điểm mạnh của Katalon so với các công cụ kiểm thử khác:

**1. Giao diện thân thiện, dễ sử dụng (so với Selenium và Appium)**:

* + Katalon cung cấp một giao diện trực quan và dễ tiếp cận, với khả năng kéo thả các đối tượng kiểm thử, giúp người dùng có thể tạo kịch bản kiểm thử mà không cần kỹ năng lập trình phức tạp. Điều này khác biệt rõ rệt so với Selenium và Appium, nơi người dùng cần viết mã từ đầu bằng Java, Python, hoặc C#.
  + Selenium và Appium là mã nguồn mở, nhưng đòi hỏi lập trình viên phải tự xây dựng framework kiểm thử riêng, điều mà Katalon đã làm sẵn.

**2. Tích hợp sẵn các tính năng đa nền tảng (so với Selenium và Appium)**:

* + Katalon tích hợp cả kiểm thử Web, Mobile, API và Desktop trong một công cụ duy nhất, trong khi với Selenium chỉ hỗ trợ kiểm thử web, và Appium chủ yếu dùng để kiểm thử di động.
  + Người dùng Selenium hoặc Appium phải tích hợp nhiều công cụ khác để thực hiện các loại kiểm thử này, làm cho quá trình trở nên phức tạp hơn.

**3. Quản lý dữ liệu kiểm thử tốt (so với TestComplete và Ranorex)**:

* + Katalon hỗ trợ mạnh mẽ cho **Data-driven testing** (kiểm thử dựa trên dữ liệu), có thể dễ dàng tích hợp và sử dụng dữ liệu từ nhiều nguồn khác nhau như Excel, CSV, hoặc database.
  + So với TestComplete và Ranorex, Katalon có quy trình quản lý dữ liệu dễ dàng hơn và trực quan hơn đối với người dùng phổ thông.

**4. Tính năng kiểm thử API mạnh mẽ (so với Selenium)**:

* + Katalon hỗ trợ kiểm thử API (RESTful và SOAP) một cách mạnh mẽ với các tính năng kiểm thử tích hợp, điều mà Selenium hoàn toàn không hỗ trợ. Người dùng Selenium cần sử dụng công cụ khác như Postman hoặc RestAssured để kiểm thử API.

**5. Tích hợp liền mạch với CI/CD (so với Ranorex)**:

* + Katalon tích hợp dễ dàng với các công cụ CI/CD như Jenkins, Bamboo, Git, Docker... giúp tự động hóa kiểm thử trong quy trình DevOps. Mặc dù Ranorex cũng hỗ trợ CI/CD nhưng thường yêu cầu nhiều cấu hình phức tạp hơn.

**6. Tài liệu và hỗ trợ cộng đồng tốt (so với Appium)**:

* + Katalon có tài liệu phong phú và một cộng đồng hỗ trợ lớn. Mặc dù Appium và Selenium cũng có cộng đồng mã nguồn mở lớn, nhưng tài liệu của Katalon được tổ chức rõ ràng hơn, phù hợp với người dùng mới.

### 2.2.2. Điểm yếu của Katalon so với các công cụ kiểm thử khác:

**1. Khả năng mở rộng và tùy chỉnh bị giới hạn (so với Selenium và Appium)**:

* + Mặc dù Katalon dễ sử dụng hơn, nhưng nó không thể tùy chỉnh sâu như Selenium hoặc Appium. Selenium và Appium cung cấp khả năng tùy chỉnh mạnh mẽ cho người lập trình viên, giúp họ kiểm thử các kịch bản phức tạp hoặc mở rộng công cụ theo nhu cầu riêng.
  + Các công cụ mã nguồn mở như Selenium có thể tích hợp với nhiều thư viện và plugin bên ngoài, trong khi Katalon bị giới hạn hơn trong vấn đề này.

**2. Hiệu suất với các dự án lớn (so với TestComplete và Ranorex)**:

* + Katalon có thể gặp vấn đề về hiệu suất khi xử lý các dự án lớn với nhiều kịch bản kiểm thử. TestComplete và Ranorex có khả năng tối ưu hóa tốt hơn khi xử lý dự án quy mô lớn và có giao diện phức tạp.
  + Mặc dù Katalon hỗ trợ nhiều nền tảng, nhưng việc quản lý kiểm thử đồng thời trên nhiều hệ thống lớn có thể gây giảm hiệu suất.

**3. Chi phí cao với các tính năng nâng cao (so với Selenium và Appium)**:

* + Mặc dù Katalon có phiên bản miễn phí, nhưng một số tính năng nâng cao, như tích hợp quản lý kiểm thử, kiểm thử bảo mật, và phân tích nâng cao, chỉ có trong phiên bản trả phí (Katalon Studio Enterprise).
  + Selenium và Appium đều là mã nguồn mở và miễn phí, giúp tiết kiệm chi phí hơn trong các dự án lớn, mặc dù cần đầu tư nhiều thời gian hơn vào thiết lập framework.

**4. Không phải là lựa chọn tốt cho kiểm thử hiệu năng (so với JMeter)**:

* + Katalon không mạnh trong kiểm thử hiệu năng (performance testing) so với các công cụ chuyên biệt như Apache JMeter. Để kiểm thử tải hoặc hiệu năng, người dùng vẫn cần sử dụng công cụ khác, trong khi JMeter hoặc LoadRunner được thiết kế chuyên biệt cho loại kiểm thử này.

**5. Cộng đồng phát triển ít hơn (so với Selenium và Appium)**:

* + Mặc dù Katalon có một cộng đồng phát triển tốt, nhưng vẫn kém phổ biến hơn nhiều so với Selenium hoặc Appium, nơi có nhiều plugin, thư viện mở rộng và tài liệu từ cộng đồng mã nguồn mở.

### 2.2.3. Kết luận:

* **Điểm mạnh của Katalon** nằm ở sự đơn giản và tích hợp sẵn các tính năng đa nền tảng, phù hợp với những dự án nhỏ và trung bình, hoặc các đội ngũ kiểm thử ít kỹ năng lập trình.
* **Điểm yếu của Katalon** là khả năng tùy chỉnh không linh hoạt bằng các công cụ mã nguồn mở như Selenium và Appium, và có thể không phù hợp với các dự án lớn yêu cầu hiệu suất cao hoặc chi phí thấp.