

BỘ CÔNG THƯƠNG

TRƯỜNG CAO ĐẲNG KỸ THUẬT

CAO THẮNG

…… 🙦  🙤 ……

ĐỒ ÁN TÌM HIỂU

KATALON

**Đề tài:**

**1.Kiểm thử phần mềm tự động-katalon**

**2.Thành viên nhóm:**

1. Lê Nguyễn Hoàng Oanh -0306221362
2. Huỳnh Nhựt Khoa -0306221342
3. Nguyễn Văn Minh Mẫn -0306221353

# Chương 1: Giới Thiệu

## Lịch ra đời

* Năm ra đời: Katalon Studio được phát hành lần đầu vào năm 2015.
* Nhà phát triển: Công ty đứng sau Katalon Studio là Katalon LLC. Đây là một công ty công nghệ có trụ sở tại Singapore, chuyên phát triển các giải pháp kiểm thử phần mềm

## Tổng quan

Katalon Studio là một công cụ kiểm thử tự động phổ biến, được đánh giá cao bởi sự thân thiện với người dùng và khả năng hỗ trợ đa nền tảng. Nó được thiết kế để đơn giản hóa quá trình tạo và thực thi các kịch bản kiểm thử tự động cho ứng dụng web, API, mobile và desktop.

Katalon Studio được xây dựng dựa trên các framework nguồn mở phổ biến là Selenium và Appium. Tuy nhiên, để tạo ra các kịch bản kiểm thử, Katalon Studio sử dụng chủ yếu hai ngôn ngữ lập trình chính là:

Groovy: Đây là ngôn ngữ lập trình chính được sử dụng trong Katalon Studio. Groovy là một ngôn ngữ lập trình động, được xây dựng trên nền tảng Java, và được thiết kế để đơn giản hóa việc viết code. Groovy cung cấp cú pháp ngắn gọn và dễ đọc, giúp người dùng dễ dàng tạo ra các kịch bản kiểm thử.

Java: Ngoài Groovy, Katalon Studio cũng hỗ trợ ngôn ngữ lập trình Java. Điều này cho phép người dùng tích hợp các thư viện Java vào các dự án của mình và sử dụng các kiến thức Java hiện có để tạo ra các kịch bản kiểm thử phức tạp hơn.

Ngôn ngữ chính là Groovy ngôn ngữ hỗ trợ là java

## Mục đích Ra Đời của Katalon Studio

Katalon Studio được ra đời với mục tiêu chính là đơn giản hóa quá trình tự động hóa kiểm thử phần mềm. Thay vì phải viết các đoạn code phức tạp bằng các ngôn ngữ lập trình như Java, Python, các kỹ sư kiểm thử có thể sử dụng Katalon Studio để tạo ra các kịch bản kiểm thử một cách trực quan và dễ dàng hơn.

Cụ thể hơn, Katalon Studio ra đời nhằm giải quyết các vấn đề sau:

* Giảm thiểu thời gian và chi phí: Tự động hóa kiểm thử giúp tiết kiệm thời gian và chi phí so với việc kiểm thử thủ công, đặc biệt là đối với các dự án lớn và phức tạp.
* Tăng độ chính xác: Các kịch bản kiểm thử được thực thi tự động giúp giảm thiểu lỗi do con người gây ra và đảm bảo độ chính xác của kết quả kiểm thử.
* Nâng cao hiệu quả: Tự động hóa kiểm thử cho phép thực hiện nhiều trường hợp kiểm thử trong thời gian ngắn, giúp tăng năng suất làm việc của đội ngũ kiểm thử.
* Hỗ trợ đa nền tảng: Katalon Studio hỗ trợ kiểm thử trên nhiều nền tảng khác nhau như web, API, mobile và desktop, giúp các doanh nghiệp mở rộng quy mô tự động hóa kiểm thử.
* Dễ sử dụng: Giao diện trực quan và thân thiện với người dùng, giúp cả những người không có nhiều kinh nghiệm về lập trình cũng có thể sử dụng được.
* Các tính năng chính của Katalon Studio giúp đạt được mục tiêu này:
* Giao diện trực quan: Người dùng có thể tạo các kịch bản kiểm thử bằng cách kéo thả các lệnh và đối tượng.
* Hỗ trợ nhiều ngôn ngữ lập trình: Ngoài giao diện trực quan, người dùng có thể tùy chỉnh các kịch bản bằng các ngôn ngữ lập trình như Groovy và Java.
* Tích hợp với các công cụ khác: Katalon Studio có thể tích hợp với các công cụ quản lý dự án, công cụ báo cáo và các công cụ CI/CD khác.
* Cộng đồng người dùng lớn: Có một cộng đồng người dùng lớn, giúp người dùng dễ dàng tìm kiếm tài liệu, hỗ trợ và chia sẻ kinh nghiệm.
* Tóm lại, Katalon Studio ra đời với mục tiêu cung cấp một giải pháp toàn diện cho việc tự động hóa kiểm thử phần mềm, giúp các doanh nghiệp tăng cường chất lượng sản phẩm, giảm thiểu rủi ro và nâng cao hiệu quả làm việc.

## Các khái niệm liên quan

* **Kiểm thử tự động (Automation testing):** Là quá trình sử dụng các công cụ và script để thực hiện các test case một cách tự động, thay thế cho việc kiểm thử thủ công.
* **Katalon Studio:** Một công cụ kiểm thử tự động mạnh mẽ, hỗ trợ đa nền tảng (web, mobile, API, desktop) và tích hợp nhiều framework phổ biến.
* **Object Repository:** Nơi lưu trữ thông tin về các đối tượng trên giao diện ứng dụng (web elements, mobile elements, API endpoints), giúp dễ dàng tìm kiếm và thao tác với chúng trong các test case.
* **Test Case:** Một tập hợp các bước thực hiện để kiểm tra một chức năng cụ thể của ứng dụng.
* **Test Suite:** Một tập hợp các test case liên quan, giúp tổ chức và quản lý các test case một cách hiệu quả.
* **Keywords:** Các hàm hoặc phương thức được định nghĩa trước để thực hiện các tác vụ cụ thể, giúp tái sử dụng code và tăng tính modular của test case.
* **Groovy:** Ngôn ngữ lập trình được sử dụng để viết các script phức tạp trong Katalon Studio.
* **Selenium:** Một framework tự động hóa kiểm thử web mã nguồn mở, là nền tảng cơ bản của Katalon Studio.
* **Appium:** Một framework tự động hóa kiểm thử ứng dụng mobile, được tích hợp sẵn trong Katalon Studio.
* **CI/CD:** Tích hợp liên tục (Continuous Integration) và triển khai liên tục (Continuous Delivery), là các thực tiễn phát triển phần mềm giúp tự động hóa quá trình xây dựng, kiểm thử và triển khai ứng dụng.

# Chương 2: Tổng quan về giao diện Katalon Studio

## Giao diện tổng quan

Giao diện của Katalon Studio được thiết kế để tối ưu hóa quy trình làm việc của người dùng, từ việc tạo test object đến việc chạy và báo cáo kết quả kiểm thử.

Cấu trúc giao diện chính. Khi mở Katalon Studio, bạn sẽ thấy các thành phần chính sau:

* + Menu bar: Chứa các menu chính như File, Edit, View, Test Cases, và các tùy chọn cấu hình khác.
  + Toolbar: Nơi tập trung các nút lệnh thường dùng để tạo, chạy, debug test case, quản lý project.
  + Project Explorer: Hiển thị cấu trúc của project, bao gồm các thư mục chứa test cases, object repository, keywords, và các file cấu hình khác.
  + Editor: Khu vực chính để viết code, tạo test case, và chỉnh sửa các đối tượng khác.
  + Log Viewer: Hiển thị các thông tin chi tiết về quá trình thực thi test case, bao gồm các log, lỗi, và cảnh báo

## Các thành phần quan trọng

* Object Repository: Nơi lưu trữ các đối tượng trên giao diện ứng dụng (web elements, mobile elements, API endpoints) để dễ dàng sử dụng lại trong các test case.
* Test Cases: Chứa các bước thực hiện cụ thể của một test case, được viết bằng ngôn ngữ Groovy hoặc sử dụng các keyword.
* Test Suites: Tập hợp các test case liên quan, giúp tổ chức và quản lý các test case một cách hiệu quả.
* Keywords: Các hàm hoặc phương thức được định nghĩa trước để thực hiện các tác vụ cụ thể, giúp tái sử dụng code và tăng tính modular của test case.

## Các tính năng nổi bật của giao diện

**Hỗ trợ nhiều ngôn ngữ lập trình:** Katalon Studio cho phép bạn viết test case bằng cả ngôn ngữ tự nhiên (keyword-driven testing) và ngôn ngữ lập trình Groovy.

**Tích hợp nhiều framework:** Công cụ này hỗ trợ các framework phổ biến như Selenium, Appium, RESTful API, và Database.

**Khả năng tùy biến cao:** Bạn có thể tùy chỉnh giao diện, cấu hình, và các plugin để phù hợp với nhu cầu của mình.

**Báo cáo chi tiết:** Katalon Studio cung cấp các báo cáo trực quan về kết quả kiểm thử, giúp bạn dễ dàng phân tích và xác định lỗi.

## Tổng kết kết giao diện

Giao diện của Katalon Studio được thiết kế để thân thiện với người dùng, ngay cả đối với những người mới bắt đầu. Với các tính năng mạnh mẽ và linh hoạt, Katalon Studio là một công cụ lý tưởng để thực hiện kiểm thử tự động cho các ứng dụng web, mobile và API.

## Ưu điểm của Katalon Studio

* **Dễ sử dụng:** Giao diện trực quan, dễ hiểu, hỗ trợ cả người dùng không có kinh nghiệm lập trình.
* **Đa nền tảng:** Hỗ trợ kiểm thử web, mobile, API, desktop.
* **Tích hợp nhiều framework:** Tích hợp sẵn Selenium, Appium và nhiều framework khác.
* **Tự động hóa mạnh mẽ:** Cho phép tạo các test case phức tạp, thực hiện các tác vụ lặp đi lặp lại.
* **Báo cáo chi tiết:** Cung cấp các báo cáo trực quan về kết quả kiểm thử.
* **Cộng đồng lớn:** Có một cộng đồng người dùng đông đảo, giúp hỗ trợ và giải đáp các vấn đề.
* **Miễn phí:** Có phiên bản miễn phí với đầy đủ các tính năng cơ bản.

## Nhược điểm của Katalon Studio

* **Khả năng tùy biến hạn chế:** So với các framework tự động hóa khác, Katalon Studio có ít tùy biến hơn.
* **Hiệu suất:** Đối với các dự án lớn và phức tạp, hiệu suất của Katalon Studio có thể bị ảnh hưởng.
* **Phụ thuộc vào công cụ:** Việc quá phụ thuộc vào một công cụ cụ thể có thể gây khó khăn khi chuyển đổi sang công cụ khác.

## Khi nào nên sử dụng Katalon Studio?

* **Dự án vừa và nhỏ:** Katalon Studio phù hợp với các dự án có quy mô vừa và nhỏ, không yêu cầu quá nhiều tùy biến.
* **Đội ngũ kiểm thử không có nhiều kinh nghiệm:** Katalon Studio giúp giảm thiểu thời gian học tập và dễ dàng sử dụng.
* **Cần một giải pháp toàn diện:** Katalon Studio cung cấp một giải pháp toàn diện cho việc kiểm thử tự động, từ web đến mobile.
* **Tổng kết:**
  + Katalon Studio là một công cụ kiểm thử tự động mạnh mẽ và dễ sử dụng, phù hợp với nhiều loại dự án. Tuy nhiên, để đưa ra quyết định cuối cùng, bạn nên cân nhắc kỹ các yếu tố như quy mô dự án, kinh nghiệm của đội ngũ, và các yêu cầu cụ thể của dự án

# Chương 3: Cách sử dụng và tính ứng dụng của phần mềm Katalon Studio

## Các tính năng chính và cách sử dụng:

1. **Record and Playback:** Ghi lại các hành động của bạn trên trình duyệt và tự động tạo ra test case.
2. **Keyword-Driven Testing:** Sử dụng các keyword có sẵn để tạo các test case một cách nhanh chóng.
3. **Data-Driven Testing:** Thực thi cùng một test case với nhiều bộ dữ liệu khác nhau.
4. **Tích hợp với CI/CD:** Kết hợp với các công cụ CI/CD để tự động hóa quá trình xây dựng và kiểm thử.
5. **Custom Keywords:** Tạo các keyword tùy chỉnh để thực hiện các tác vụ phức tạp.
6. **Parallel Testing:** Thực thi nhiều test case đồng thời để tiết kiệm thời gian.

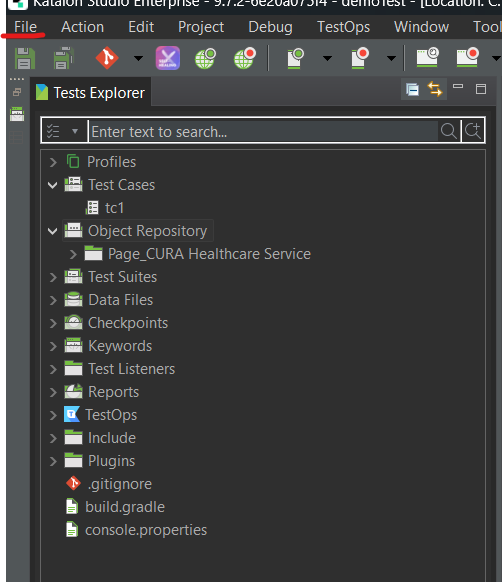
## Tính ứng dụng của Katalon Studio

1. Katalon Studio được ứng dụng rộng rãi trong nhiều lĩnh vực, từ các dự án nhỏ đến các dự án lớn, phức tạp. Dưới đây là một số ứng dụng điển hình:
2. **Kiểm thử ứng dụng web:** Kiểm tra chức năng, hiệu suất, giao diện người dùng của các ứng dụng web.
3. **Kiểm thử ứng dụng di động:** Kiểm thử các ứng dụng trên các nền tảng di động như iOS và Android.
4. **Kiểm thử API:** Kiểm tra các API để đảm bảo chúng hoạt động đúng.
5. **Kiểm thử ứng dụng desktop:** Kiểm thử các ứng dụng chạy trên máy tính để bàn.
6. **Kiểm thử hồi quy:** Kiểm tra xem các thay đổi mới có ảnh hưởng đến các chức năng đã tồn tại hay không.
7. **Kiểm thử khói:** Kiểm tra các chức năng cốt lõi của ứng dụng sau khi triển khai.

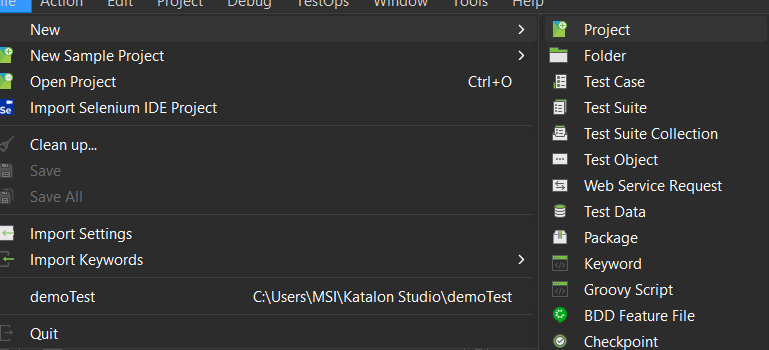
## Cách sử dụng Katalon Studio

Katalon Studio được thiết kế với giao diện trực quan, dễ sử dụng, giúp người dùng tạo và thực thi các test case một cách nhanh chóng và hiệu quả. Dưới đây là các bước cơ bản khi làm việc với Katalon Studio:

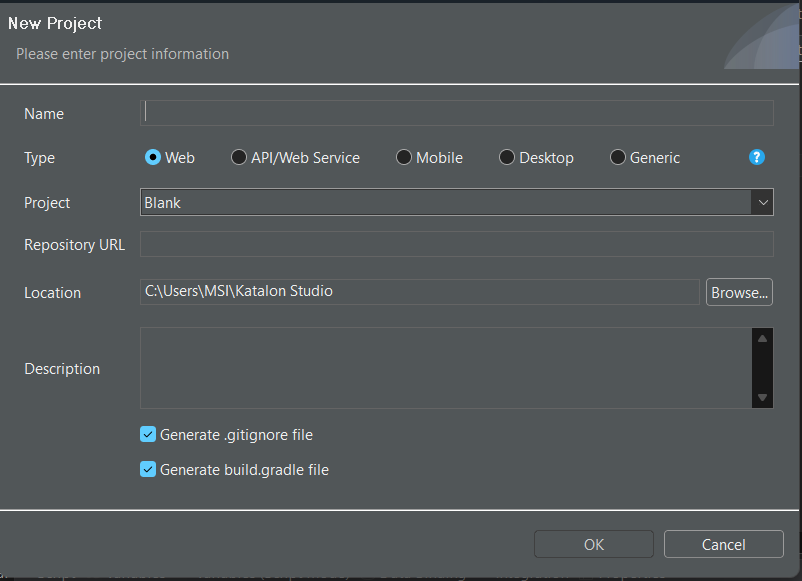
1. **Tạo Project:** Tạo một project mới để quản lý các test case, object repository và các tài nguyên khác.



Hình. 1.Ấn chọn file trên thank tapbar

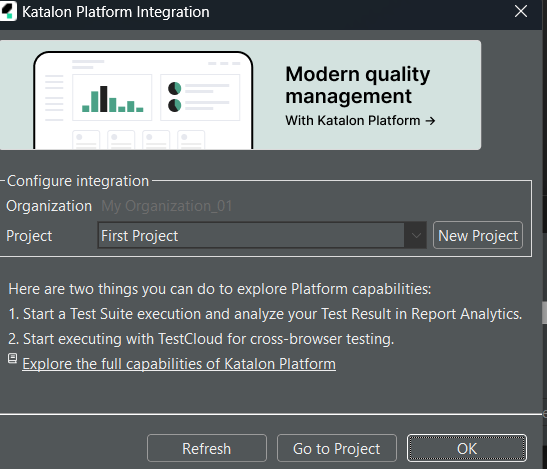


Hình. 2.Ấn chọn new-project



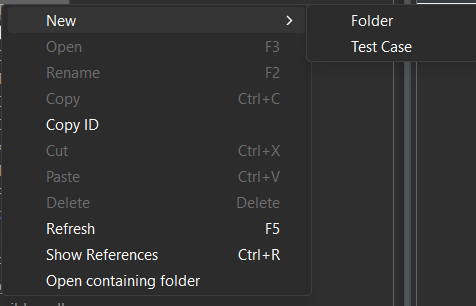
Hình. 3.Đặt tên cho project

Đặt tên cho project và lựa con các type tùy thuộc vào dự án mà bạn muốn thực hiện

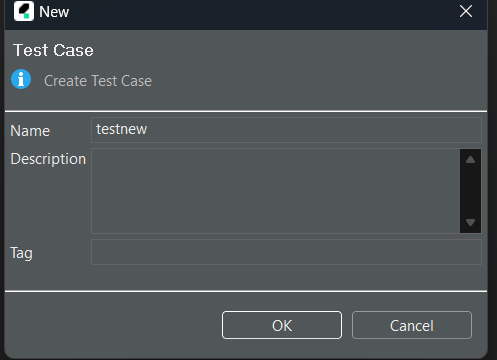


Hình. 4. Bắt đầu dự án

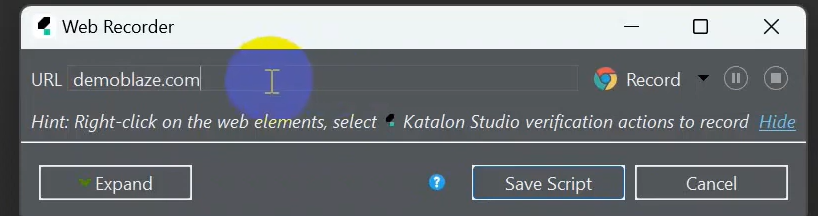
2. **Tạo 1 test case**



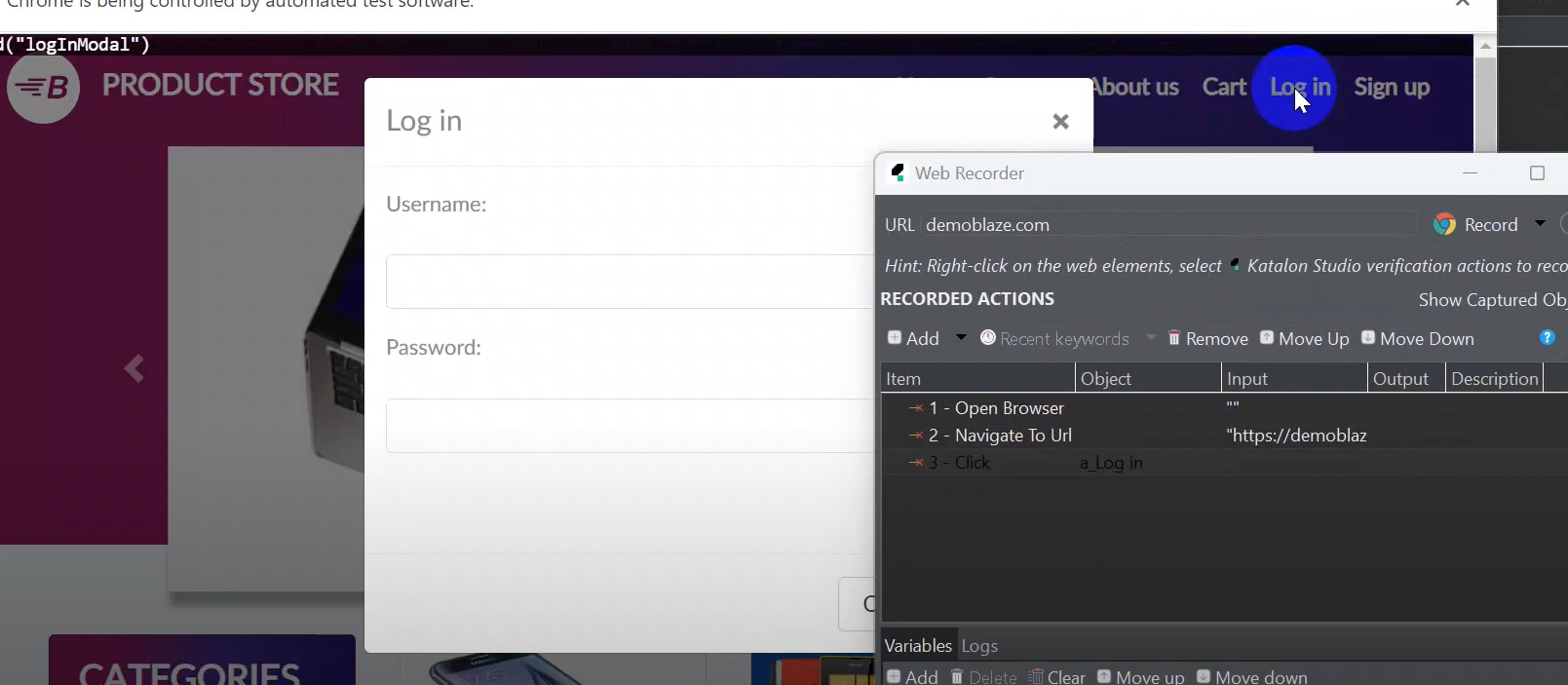
Hình. 5.Tạo Test case mới



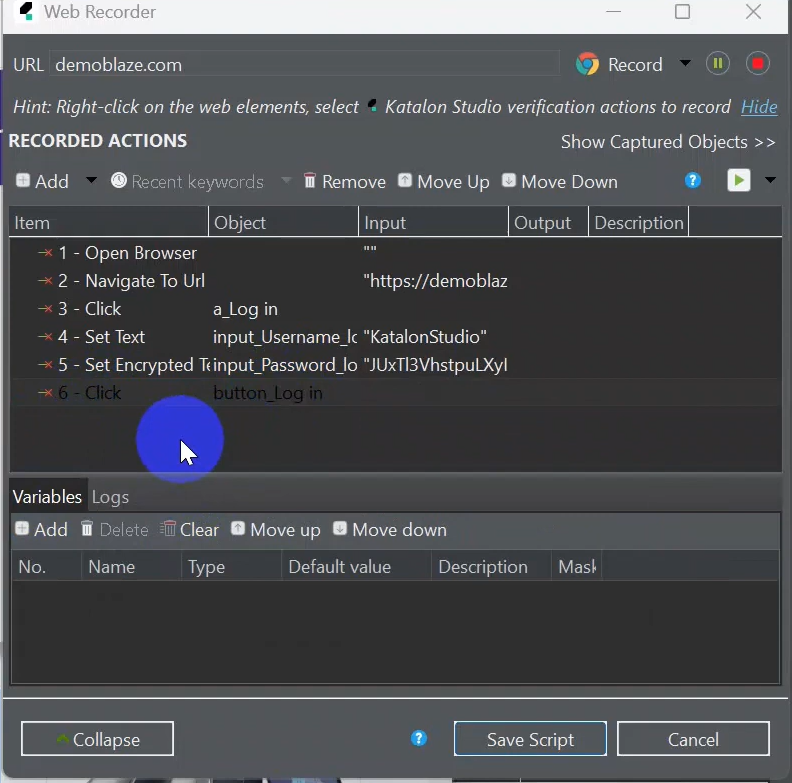
Hình. 6.Đặt tên test case



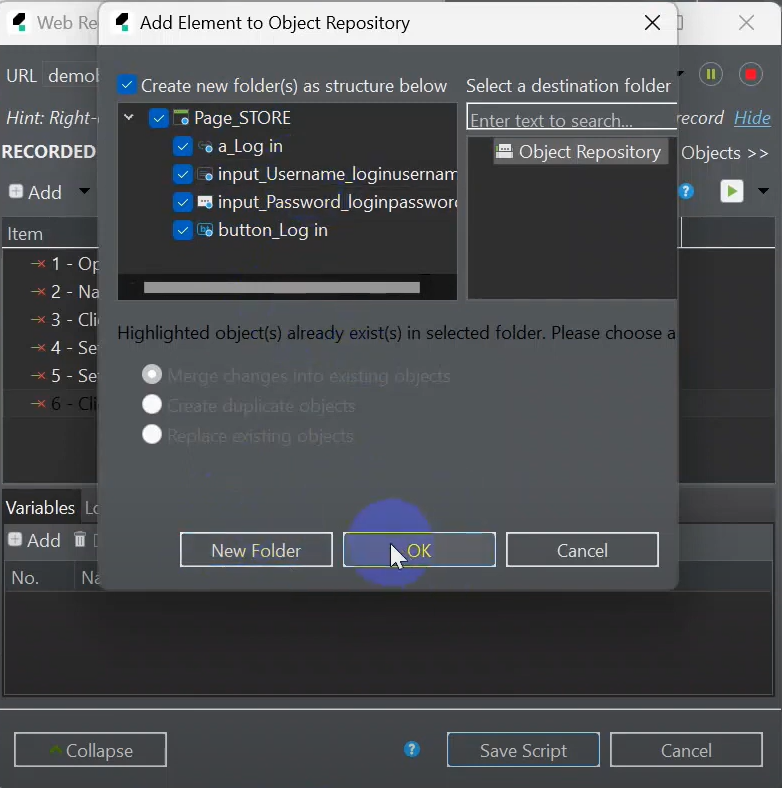
Hình. 7.truy cập truy wed và record theo tac



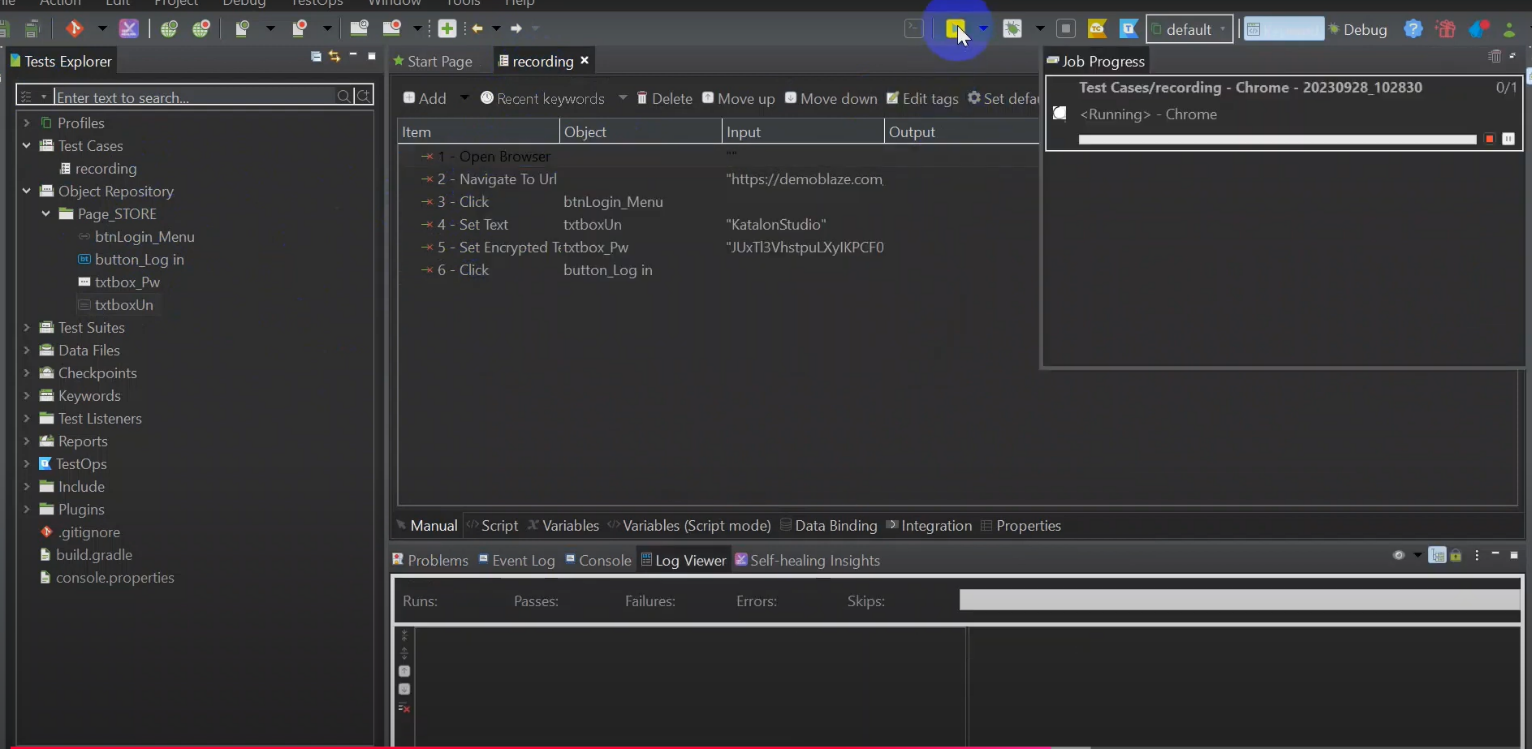
Hình. 8.Thực Hiện các thao tắc đăng nhập trên wed

Thực hiện các thao tắc tên trang WED các thao tắc sẽ được lưu lại 

Hình. 9.Các acction đã thực hiện

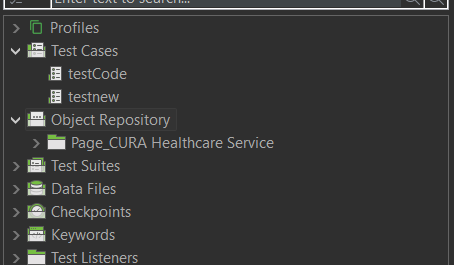


Hình. 10.Lưu các trường đã thực hiện



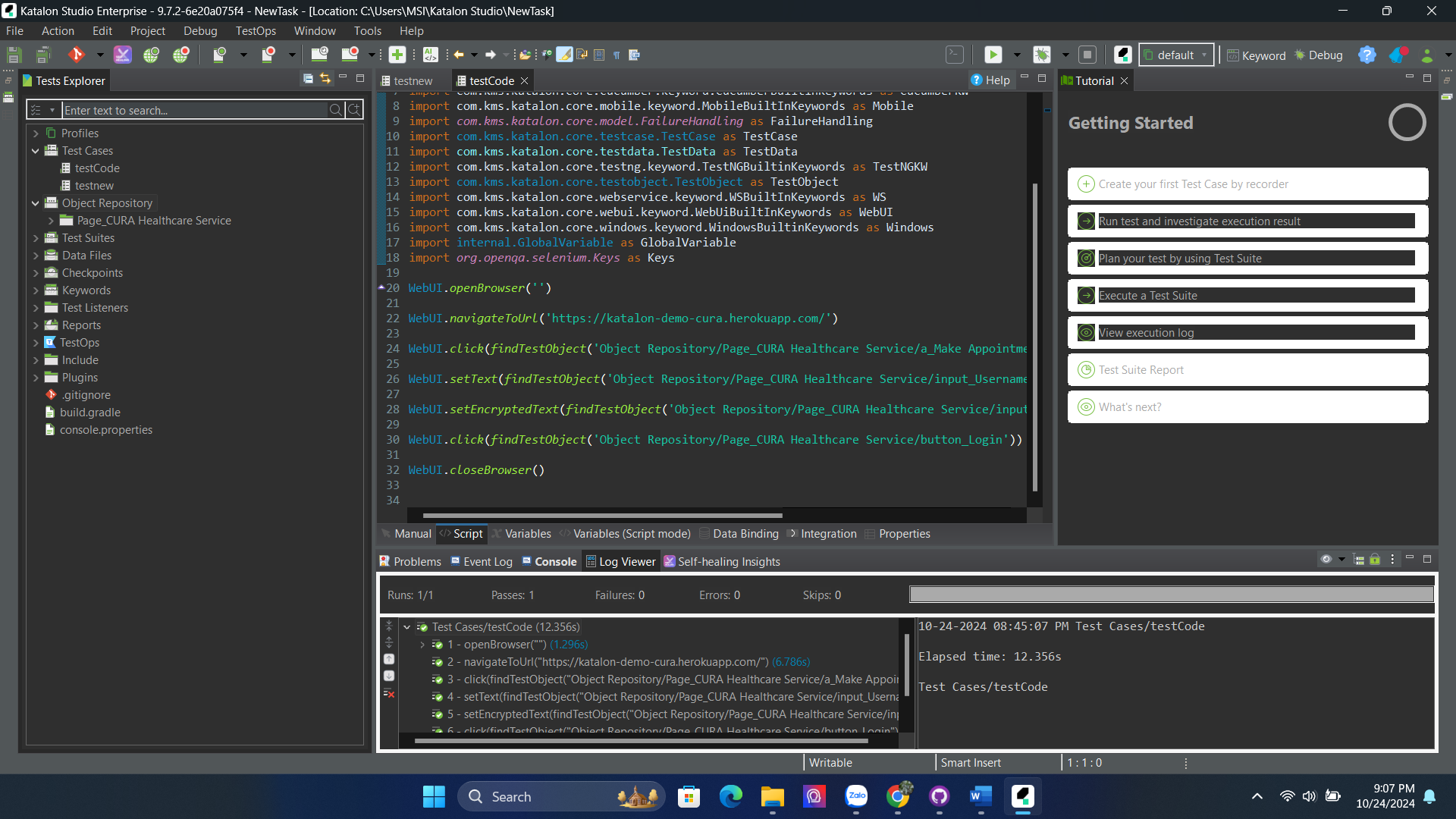
Hình. 11.Thực hiện chạy lại các theo tác

**3. Tạo Object Repository:** Sử dụng Spy Web để xác định các element trên giao diện web và lưu trữ chúng vào Object Repository để tái sử dụng trong các test case.



Hình. 12.Tạo Object Repository

Object Repository được tạo ra mặt định trong quá trình record , có thể tạo thêm Object Repository để cho quá trình test case có thể thuận tiện hiện hơn

**4. Tạo Test Case Groovy:** Sử dụng ngôn ngữ Groovy để viết các bước thực hiện của test case.

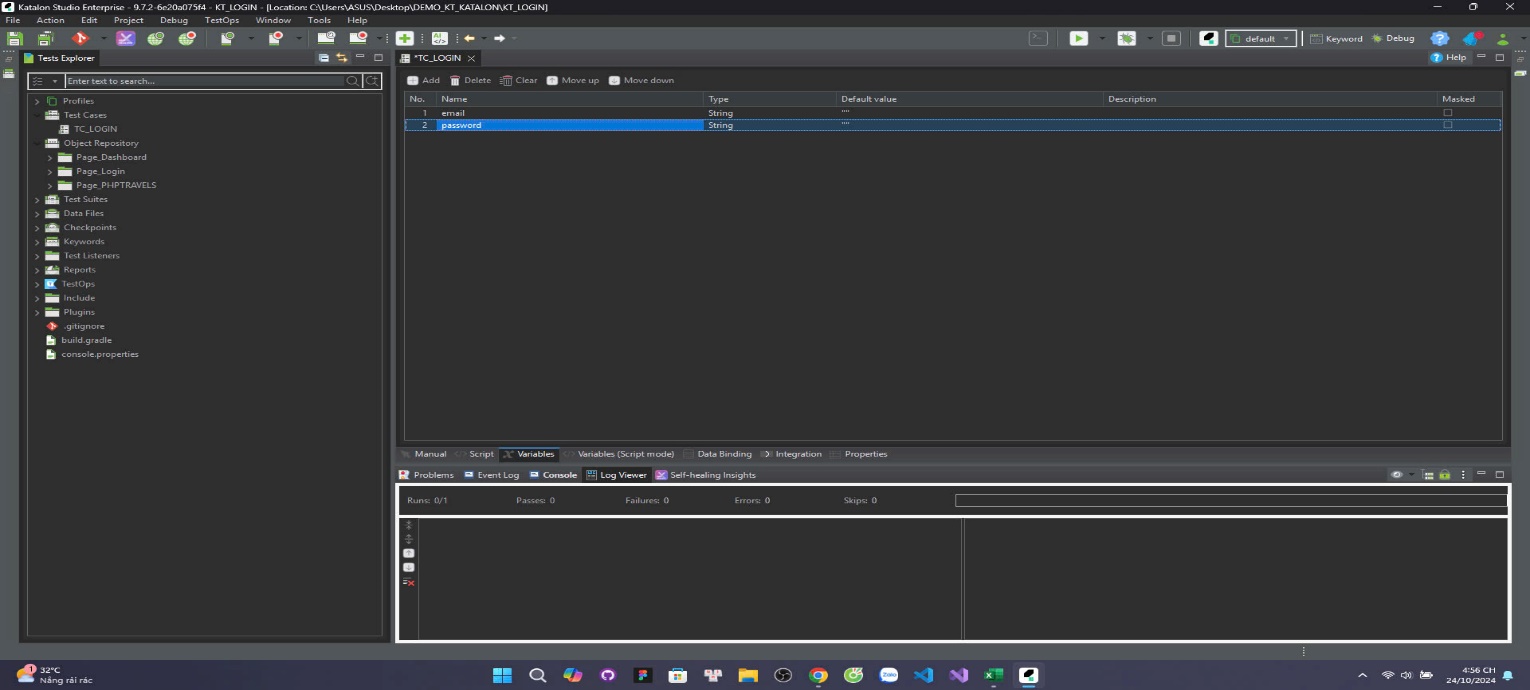
Hình. 13.Test case với ngôn ngữ Groovy

Groovy là một ngôn ngữ động, cho phép lập trình linh hoạt với khả năng điều chỉnh mã kiểm thử một cách dễ dàng mà không cần biên dịch lại toàn bộ dự án

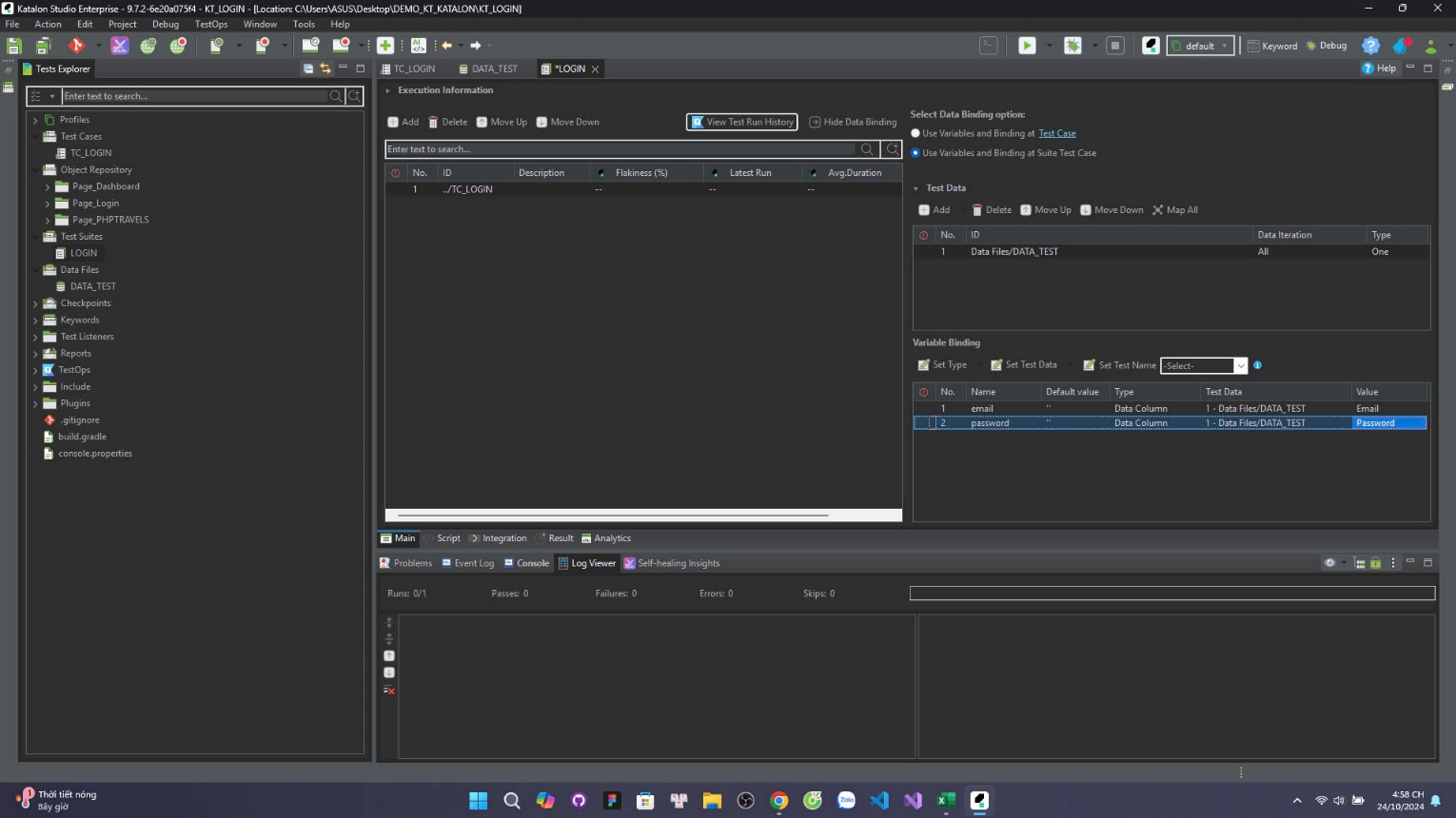
5.**Cách sử dụng tổng quan cơ bản**

**Thức hiện các bước từ hình 1 đến hình 10 để bắt đầu tạo testcase cơ bản**

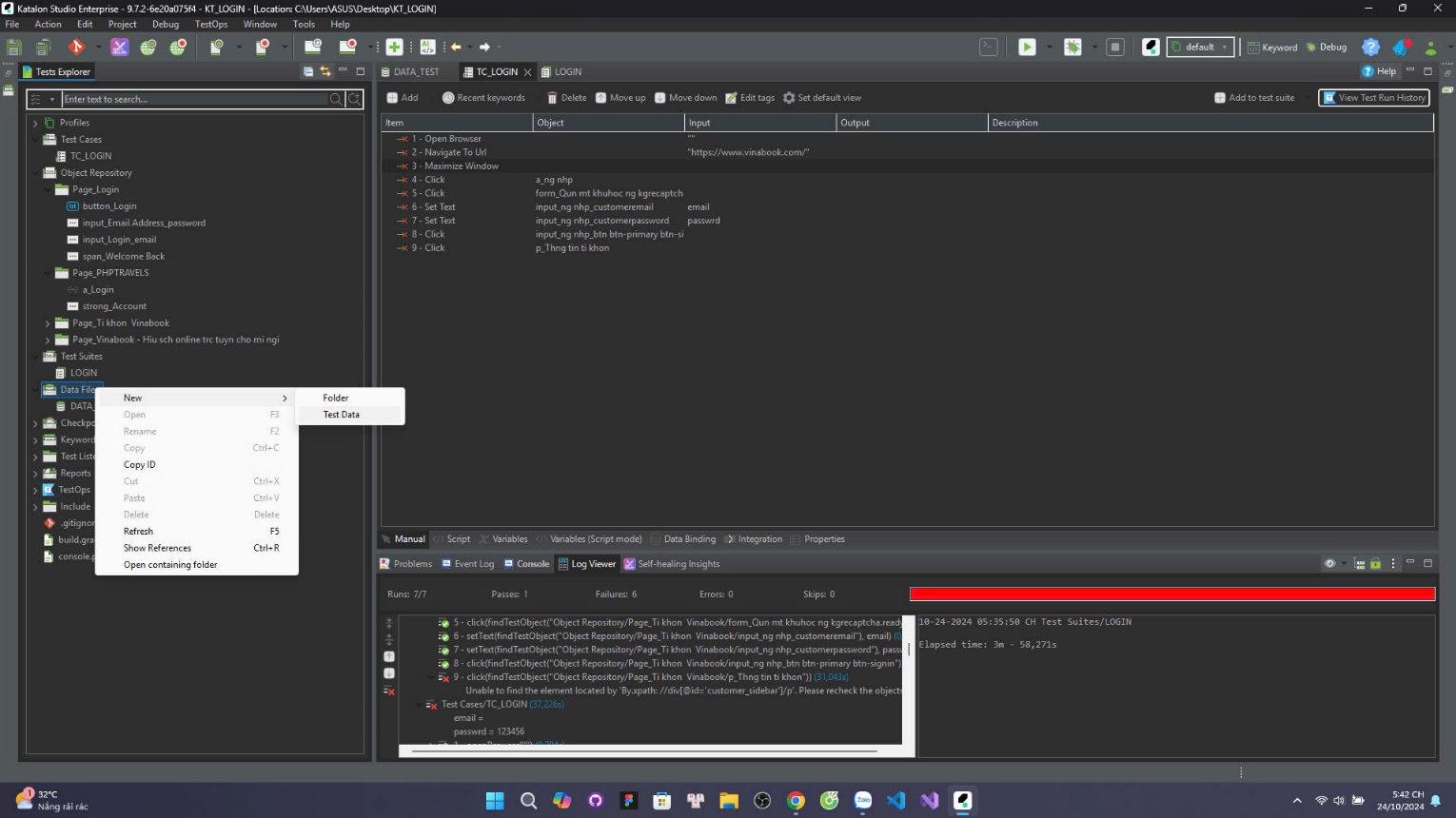
Có thể tạo các biến để truyền vào để tối ưu cho các test case cần thiết



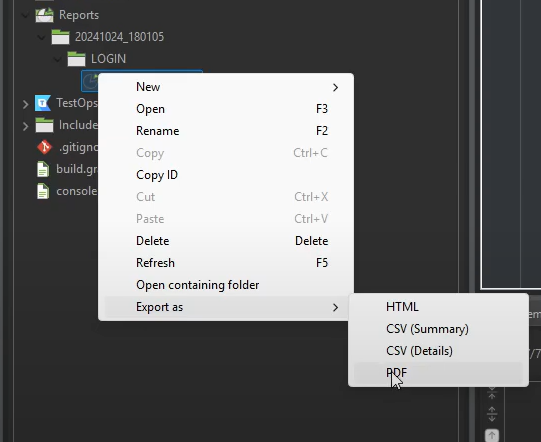
Hình. 14.Tạo biến động để truyền testcase từ bên ngoài vào



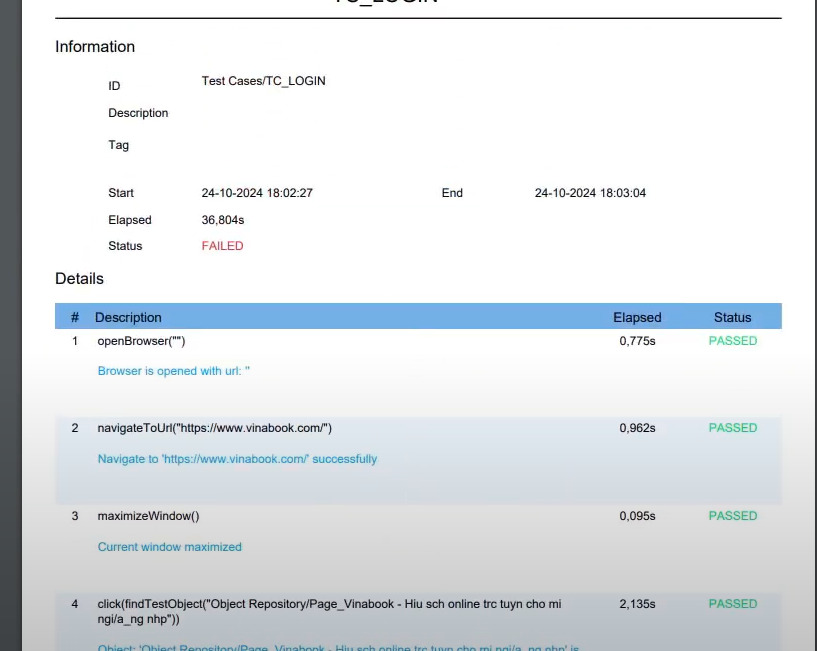
Hình. 15.Tạo test suite để chọn dữ liệu trong file vừa import



Hình. 16.Tạo test suite để chọn dữ liệu trong file vừa import



Hình. 17.In báo cáo các test case thực hiện



Hình. 18.Thông tín các test case đã test