CSCE 5430.003

Sahi Software

Group 11- Project 3 Testing Report

Overview

Sahi is a software testing tool that provides a range of features to support testing.

General Features:

- Sahi has Simple & Powerful APIs to easily identify elements on DOM, perform mouse, keyboard, and touch actions seamlessly.
- Sahi embeds automatic waits and eliminates wait for statements and can navigate inconsistent page loads.
- Sahi Object Spy and Recorder to identify elements across devices and software. It can also work on applications that generate dynamic IDs for elements.
- Sahi implements Business friendly frameworks and uses inbuilt Business Driven Test Automation to let business analysts and non-technical professionals contribute towards test cases.
- Sahi employs automatic logging while automating applications that contains the complete information about the run. This helps testers identify the exact line where script is failing. All logs are stored in database.
- Sahi can run thousands of scripts that are packed into a suite on a single machine in parallel mode or can be distributed across machines.

Some of the notable features include:

- Record and Playback: Sahi Pro allows testers to record their interactions with the application and play them back to create automation scripts.
- Multi-Browser Support: Sahi Pro supports multiple browsers, including Chrome, Firefox, Safari, and Internet Explorer.
- Parallel Execution: Sahi Pro allows tests to be run in parallel across multiple machines, reducing the time required for testing.
- Script less Automation: Sahi Pro supports script less automation, allowing testers to create automation scripts without any programming knowledge.

Roles and Input Combinations

- Sahi is suitable for use by a range of stakeholders involved in software testing, including developers, testers, and QA engineers.
- Sahi uses control + hover combination to identify elements in DOM. Uses Sahi Script to store the user events and is an extension of JavaScript, capable of interacting with browser efficiently and perform reads on file system, access database and can call Java etc.,
- Sahi Pro offers a variety of ways to identify web elements on a page, including through traditional methods such as ID, name, and CSS selectors, as well as advanced

techniques like relation APIs (such as leftOf, under ,near, in, and rightOf) that allow testers to identify elements with respect to each other.

- Sahi is Browser and Operating System independent in most cases. Javascript is injected into webpages using proxy. In 95% of the test cases, this will work. Sahi falls back to native events, when Javascript events may not work. These require focus on test window and prevents parallel testing.
- Sahi can be used to test Web, Mobile, Desktop and SAP applications. It is used in automating test cases.

Sahi Economics

Sahi started as Open-Source project in 2005 with focus on automation of emerging web 2.0 technologies. Sahi evolved into Sahi Pro that handles automation on modern web browsers, mobiles, and desktop applications.

Sahi Pro has 4 pricing plans available for customers:

- Sahi Pro for Web
 - Works across browsers and operating systems
- Sahi Pro for Desktop
 - Works for windows desktop applications (WPF, .NET, etc.,)
 - Java based applications.
- Sahi Pro for Mobile
 - Mobile native & hybrid applications in iOS and Android
- Sahi Pro for SAP
 - SAP GUI for windows

All plans support Database, File system, REST/Web services. Each plan has user / concurrent licenses available. Sahi Pro is used by over 1000 companies globally, including Fortune 500 companies, government organizations, and startups. Some of the notable users of Sahi Pro include Accenture, Cognizant, IBM, Infosys, and Wipro. Sahi Pro offers 30 days free trial. Sahi reviews and testimonials are strong according to the official website. Product selling point is the Record and playback tool functionality which works across all major browsers regardless of their versions.

History

Sahi is an open-source test automation tool for web applications. It was initially created in 2005 with a specific focus on the automation of emerging web 2.0 technologies aimed at testers. Over the years, the tool has undergone significant development, and Sahi Pro is now an enterprise-grade test automation platform used by over 1000 companies worldwide.

Testing Interface

The Sahi Pro user interface is designed to be simple and intuitive. The testing interface consists of several components, including the Sahi Dashboard, Test Suite Editor, Test Runner, and Test Logs.

Sahi Dashboard: Sahi Dashboard consists of available browsers on the system and include slinks to configure, Scripts, script editor, SSL, Docs, Logs, Bin

Configure: Configure window contains Sahi Pro configurations like browser types, Access Control, URL history. Each configuration details are present at the top of the input window. User can click save to save the changed configuration.





Scripts: clicking on scripts will launch file explorer on windows to show the recorded scripts stored in .sah format.

scripts	× +			- 0	×
🕂 New ~		∿ Sort ~ 🗮 View ~	<u>چ</u>		
\leftrightarrow \rightarrow \checkmark \uparrow	« SahiPro > userdata > scripts >	∽ C Search script			م
合 Home	^ Name	Date modified	Туре	Size	
🔉 📥 Naga Vara Prade	🗖 demo	15-04-2023 06:25 PM	File folder		
> 📥 OneDrive - UNT	🗖 java	15-04-2023 06:25 PM	File folder		
🔀 Gallery	apis	15-04-2023 06:25 PM	SAH File	401 KB	
I	first_recording	20-04-2023 06:59 PM	SAH File	1 KB	
🔜 Desktop 🛛 🖈 🖡	global_include	15-04-2023 06:25 PM	SAH File	3 KB	
🚽 Downloads 🖈	new_recording	16-04-2023 04:14 PM	SAH File	1 KB	
📕 Documents 🖈	polymer-apis	15-04-2023 06:25 PM	SAH File	224 KB	
🔀 Pictures 🛛 🖈					
🕗 Music 🛛 🖈					
🗾 Videos 🛷					

User can review the recorded steps using a text editor like Visual Studio Code



Script Editor: Clicking on Editor opens web browser and localhost:9999 is opened. Script editor contains file navigator, edit window, action buttons at the top to work on Sahi Scripts. Users can segregate script suite, playback written script, and debug sah script.

🜌 Editor - new_recording.sah 🛛 🗙	+	~ -	0)	×
$\leftarrow \rightarrow C$ (0 localhost:9999/_s	i_/spr/editor/editor.html	📃 👽 e 🚸 🗯			
📑 New 👻 🖺 Open 🔛 Save 👻 😫	🖞 Create Suite 💌 📕 Create AR 🦹 Find Usage 👙 Playback 📕 Stop All 🗙 Clear Break Points 🛛 More 💌 💠 😅		Docs	Logs	
Navigator 🔛 🚍 🛸 🖏 «	Kictions.sah & Kousing_csv.sah & Shoppingcart.dd.csv & training.suite & 4_datadrive1.sah & artifact_resource.sah & polymer-apis.sah & Kousing_csv.sah & Kous	new_recording.sah c>	•	×	* >>
×	🍕 🧕 Auto Format 🖡 Create Function 🏚 Show Context		1 Viev	w Doc	
 Scripts (7) C) Geno (264) C) Java (3) B) asis.sah C) First_recording.sah C) pola_include.sah D) Polymer-apis.sah D) polymer-apis.sah 	<pre>1 LsetValue(_textbox("user"), "mstY; 2 setVasword(_password"), "MgkKEQ8U"); 3 _clck(_submit("Login")); 4</pre>				

Docs opens Sahi Documentation.

Introduction Using Sahi S	ahi APIs Sahi Scripting F.A.Q. Trouble Shooting (View Online Docs)	۹ ۴
	Introduction	
Sahi Pro Entroduction Sahi Pro Editions Main Features Supported Features and	Sahi Pro v9 7.0 is now available! See what's new in Sahi Pro v9 7.0!	
Technologies Browser Support Sahi Name What's paw?	Sahi Pro is a suite of mature, business-ready tools for automated testing of Web, Webservices, Mobile, Windows Desktop, SAP GUI and Java applications. For testing teams which need rapid and reliable automation, Sahi Pro would be the best choice among automation tools.	
Sahi Pro Desktop Add-on Sahi Pro Mobile Add-on Sahi Pro SAP Add-on	Sahi works well in Agile development environments, enabling rapid automation and maintenance with easy integration with build systems. Sahi saves time and effort with faster development, less maintenance and fast distributed playback.	
Sahi Pro Runner Sahi Pro Starter Sahi License Hub Architecture Sahi Script	Sahi Pro Editions	
 Sahi OS Sahi Pro Other Versions 	Sahi Pro	
	Sahi Pro helps automate functional testing of web applications. Sahi Pro by default supports web application and REST API automation. Sahi Pro is well suited for cross- browser/multi-browser testing of complex web 2.0 applications with lots of AJAX and dynamic content. Sahi Pro runs on any modern browser which supports javascript.	
	Sahi Pro Desktop Add-on	
	Sahi Pro Desktop Add-on adds support for Windows desktop applications (Win32, WPF, WinForms, NET, SilverLight, XBAP, Universal Windows Platform (UWP)), Java Desktop Applications, Java Applets and Java web start applications. It has been introduced as an Add-on since Sahi Pro v7.0.0.	
	Sahi Pro Desktop Add-on license needs to be purchased separately	
	If you want to automate only Desktop applications (and not web applications), you can purchase and use Desktop Add-on license alone. No need to purchase Sahi Pro license.	szatri,

Logs list all the runs completed and shows each action run time in milliseconds.

🜌 Suites List 🛛 🗙 🕂		v ·	- o ×
$\leftarrow \rightarrow$ C () localhost:9999/_s_/dyn/pro/DBRe		년 🏚 🖷 👽 🖉 💶	• 🛛 🍪 E
Suites		Script	Wise Summary
Delete Compare Consolidate Report	Show Graphs	Show Testcase Summary Configure columns Sync Filter Reset filters Download as	Download
50 V H 4 Page 1 of 1 ▶ H Ŏ		Displaying	1 to 1 of 1 items
	P	F E Sk All	Choose 🗸
Script Name Suite Name Base Url	Browser Total Scripts	Status Summary	/ Status
sirst_recording.sah	edgenew 1	Passed: 1	SUCCESS

Enabling Traffic logs lists all the sites visited during the automation process.

logs	× +			- 0 X
🕂 New ~ 🐰		Ĵ ↑↓ Sort ~ 🗮 View ~	<i>©</i>	
$\leftrightarrow \rightarrow \checkmark \uparrow$	≪ SahiPro → userdata → logs	✓ C Search logs		م
合 Home	Name	Date modified	Туре	Size
🔉 📥 Naga Vara Prade	🗖 playback	15-04-2023 06:26 PM	File folder	
> 📥 OneDrive - UNT	📄 sahi0	22-04-2023 05:53 PM	Text Document	0 КВ
🔀 Gallery	sahi0.log.lck	22-04-2023 04:58 PM	LCK File	0 КВ
I	📄 sahi1	22-04-2023 04:58 PM	Text Document	4 KB
🗖 Desktop 🛷 🖡	📄 sahi2	22-04-2023 04:56 PM	Text Document	4 KB
🚽 Downloads 🖈	📄 sahi3	20-04-2023 07:22 PM	Text Document	12 KB
📕 Documents 🖈	📄 sahi4	20-04-2023 06:35 PM	Text Document	3 KB
🔀 Pictures 🛛 🖈	📄 sahi5	16-04-2023 04:13 PM	Text Document	4 KB
🕑 Music 🛛 🖈	📄 sahi6	16-04-2023 02:17 PM	Text Document	4 KB
Videos 🖈 11 items 1 item selected	2.30 KB	15_01_2022 06.50 DM	Tevt Nocument	



Users can use Configure License button to add license and reset product keys.

Sahi Pro			×
		Sahi Pro Capabilities Enabled	
	Browser	Available	
	📃 🗌 Desktop	Not Installed	
	Mobile	Not Installed	
	SAP	Not Installed	
		Reset Product Key	
		OK Cancel	

Clicking browser name will open the Sahi Start window in the selected browser. User can now open Sahi Controller to start recording the automation steps using Alt key + Double click on page.



Sahi Controller can be used in two modes, 1) Sahi and 2) Java. In Sahi mode Controller contains Record, Playback, Clipboard, and Info tabs used to automate. User can click on red button at top to start automation. Recorded steps are show in Evaluate expression window. User can again click on red button to stop the recording. Clicking the settings icon shows the Script Directory, Script Name, AR file to save the recorded steps to a .sah file.

Record	Playback	Clipboard	Info	Sahi Pro 9.7.0
Click 🔵 t	o start reco	rding.		• •
Script Di	rectory:	:/SahiPro/use	erdata/scrip	ots/ 🗸
Script Na	ame: fi	rst_recording.	sah	
AR File:				🛸 🗆

Automation using Sahi:

Let us start by recording and playing back a simple script. Sahi Pro is not just a record and playback tool, recording is a steppingstone to creating automation scripts. User can start application by clicking the application shortcut on desktop. Let's record our first script. Enter a script name to save the recording. And click record button.

🜌 Sahi Start 🛛 🗙	+	× -	Sahi Controller - Google Chrome — — — — — — — — — — — — — — — — — — —
← → C a sahi.example.co	m/_s_/dyn/Driver_initialized?browserType=chrome#	🖈 🗖 😸 Incognit	Record Playback Clipboard Info Sahi Pro 9,7.0 Click • to start recording. • • Script Directory: C:/SahiPro/userdata/scripts/ • Script Name: • • AR File: • •
	Press the ALT key and Double Click on page to bring up the Sahi Controller. Sahi Scripts can be recorded and played back from the Controller. Enter start URL: [http://sahilest.com/demo/training Go		Press CTRL key and hover mouse over any element. Accessor: div("content")
	□ Change Resolution □ Change User Agent Logs Online Documentation Test Pages Sample Application Controller mode: ● Sahi ○ Java		Assert List Properties Diagnose Docs Editor
	Tyto Software		Sahl-Browser V - Append HighlightChoose V History

Record	Playback	Clipboard	Info		Sahi Pro 9.7.0	
Click ● t	o start reco	rding.			• •	
Script D	irectory:	C:/SahiPro/use	erdata/s	cripts/	~	
Script N	ame: s	ahi_sample_ru	un.sah			
AR File:					🔹 🗆	
Record	Playbac	k Clipboa	ard I	nfo	Sahi Pro	9.7.0
Record	ing in sahi <u>.</u>	_sample_ru	n.sah		٠	٥

- Ener a start URL: <u>http://sahitest.com/demo/training</u> to open Sahi provided demo application to test.
- Click on Go button to start recording steps in application.
- User can see the navigateTo expression being recorded onto script.
- When user types in username, the event is recorded.
- All the steps are autosaved to the file.
- User now clicks on login button, to record the login action.

Enter start URL: http://sahitest.com/demo/training Go

Evaluate expression	Recorded steps	Diagnose Docs Editor
_navigateTo("http://sahite	est.com/demo/training")); ^
Sahi Training S Username test Password Login Not a user? <u>Regi</u>	ite	Press CTRL key and hover mouse over any element. Accessor: textbox("_sahi_ignore_url") textbox("_sahi_ignore_url") Hiligh Value: http://sahitest.com/demo/training Set Prefix: Hove Mode: BROWSER Session key: sessionId Assert List Properties Other Actions: Choose APIs List:
Use test/secret to I	ogin	Evaluate expression Recorded steps Diagnose Docs E _setValue(_textbox("user"), "test");
Sahi Training Username test Password Login Not a user? <u>Re</u>	; Site	Press CTRL key and hover mouse over any element. Accessor: •• _textbox("_sahi_ignore_url") Alternatives: _textbox("_sahi_ignore_url") Value: http://sahitest.com/demo/training Prefix: Mode: BROWSER Session key: session Assert List Properties APIs List:
Use test/secret to	o login	_navigateTo("http://sahitest.com/demo/training"); _setValue(_textbox("user"), "test"); _setPassword(_password("password"), "MgkKEQBU");
All availa Title In stock Cos Core Java 5 Rs. Ruby for Rails 12 Rs. Python 7 Rs. Cookbook 7 Add Clee	ble books t Add quantity to cart 300 0 200 0 350 0 ar Logout	Recording in sahl_sample_run.sah Press CTRL key and hover mouse over any element. Accessor: (*) textbox("_sahl_ignore_url") Alternatives: textbox("_sahl_ignore_url") Value: http://sahlest.com/demo/training Prefix: BROWSER Session key: session1d Assert List Properties Other Actions: - Choose APIs List:
My C Title Quantity Grand Total: 0	Cart Unit Cost Total Cost	Sahl-Browser ♥ → Append HighlightChoose ♥ ♥ History

Now we add books quantity to cart.

	All a	wailable b	ooks
Title	In stock	Cost	Add quantity to cart
Core Java	5	Rs. 300	1
Ruby for Rails	12	Rs. 200	2
Python Cookbook	7	Rs. 350	2
Cookbook	Add	I Clear Lo	gout

All the actions are recorded in Recorded steps view.

_click(_submit("Login")); _setValue(_textbox("q"), "1"); _setValue(_textbox("q[1]"), "2"); _setValue(_textbox("q[2]"), "2"); _click(_button("Add"));

After Add button is clicked, books are added to the Cart.

	Μ	ly Cart	
Title	Quantity	Unit Cost	Total Cost
Core Java	1	Rs.300	Rs.300
Ruby for Rails	2	Rs.200	Rs.400
Python Cookbook	2	Rs.350	Rs.700

In this test case we verify that the Grand Total is correct, for the books quantity we added to cart. We identify the total element and create an assertion for it.

To do that we need to follow the step shown in the controller.

Press Control key and hover mouse over total element.

Press	CTRL	kev	and	hover	mouse	over	anv	element.
					1110 000	· · · ·	· · · · · ·	0.0

Press CTRL key and hover mouse over any element.								
Accessor: 🛶 _textbox("total")								
Alternatives: _textbox("total")								
Value: 1400								
Prefix:				Hover				
Mode:	BROWSER	Session key:	sessionId					
Assert List	Properties 📑	Other Actions:	Choose	~				
APIs List:								

Users can now click on the Assert button to get different assert suggestions that can be done on the selected element.

Assert List Properties 🔂 Other Actions: Choose	~
APIs List:	
Evaluate expression Recorded steps Diagnose Docs	<u>Editor</u>
_assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));	•

Generated assert statements check if the element is existing in DOM, visible to user and verifies if it is equal to 1400 value. User can verify the assertions by clicking the arrow button below the Evaluate expression tab.

Evaluate expression			Recorde	d steps		<u>Diagnose</u>	<u>Docs</u>	<u>Editor</u>
_assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));								
Sahi-Browse	r 🗸	\rightarrow	Append	Highlig	ht	Choose 🗸		<u>History</u>
true								
	li di seconda di second							

Changing 1400 value to 1450 and then evaluating the asserts will result in test failure.

Evaluate expression	Recorded steps	Diagnose	Docs Editor				
_assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1450", _getValue(_textbox("total")));							
Sahi-Browser \checkmark \rightarrow	Append Highlig	ihtChoose 🗸	✓ <u>History</u>				
[Assertion Failed] Expected:"1450" Actual:"1400"			li				

When we are good with the assert statements, clicking on Append button appends the generated asserts to the script.

Evaluate expression	Recorded steps	<u>Diagnose</u>	Docs Editor			
<pre>_assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));</pre>						
Sahi-Browser ∨ →	Append Highligh	ntChoose 🗸	✓ <u>History</u>			
Appended:						
_assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));						

Actions that we perform on the browser are automatically recorded. Actions that we perform on the Sahi Controller, need to be appended to the script manually.

Sahi Training Site Username Password Login Not a user? <u>Register</u>	Press CTRL key and hover mouse over any element. Accessor: textbox("total") Alternatives: _textbox("total") Value: 1400 Prefix:	Click Hilight Set Hover
Use test/secret to login	Evaluate expression Recorded steps Diagnose D _click(_button("Logout"));	<u>)ocs Edi</u>

Let's logout, examine the recorded script and run the automation from start.

Clicking red button will stop the recording and save all recorded steps to the .sah file provided at the start of recording.

Click on Editor to open Sahi Scripts Editor. This opens browser and reaches localhost:9999 website to open editor hosted locally by Sahi.

Evaluate expression	Recorded steps	<u>Diagnose</u>	<u>Docs</u>	<u>Editor</u>
click(button("Loaout"))	:			

User can see the recorded steps while automating web application. All steps are recorded in JavaScript statements. Main advantage is user does not have to deal with xpaths or CSS selectors to identify an element on DOM. Wait statements are also taken care implicitly by Sahi.

Z Editor - sahi_sample_run.sah 🗙	+	~ - 0 X
$\leftarrow \rightarrow \mathbf{C}$ (0 localhost:9999/_:	s /spr/editor/editor.html?dir=C%3A%2FSahiPro%2Fuserdata%2Fscripts%2F&file=sahi_sample_run.sah 🖻 🛧	
🕒 New 🔻 🖺 Open 🔛 Save 🔻 🎑	🖞 Create Suite 🔻 📕 Create AR 🕌 Find Usage 🔅 Playback 📕 Stop All 🗙 Clear Break Points 🛛 More 💌 💠 🖒	Docs Logs 🌌
Navigator 🗄 🖻 🛱 🛠	sahi_sample_run.sah c ×	🗔 🗙 🗝
✓	🔍 🎽 Auto Format 🗚 Create Function 🏂 Show Context	🎬 View Doc 🚺
 Cripts (9) Cripts (9) Cripts (2) Apix.sah First_recording.sah Global_include.sah Global_include.sah Cripts.sah Rew_recording.sah Rew_recording.sah Sahi_sample_run.sah 	<pre></pre>	 Context

Let's playback the script that we just recorded. Open Sahi Controller, and go to Playback tab.

Record	Playback	Clipboard	Info			Sahi Pro	9.7.0
Set script from file:							
Script Dire	ectory:	C:/SahiPro/use	erdata/s	cripts/			~
File:	6	ahi_sample_ru	ın.sah				
🗹 Start URL:		sahi_sample_run.sah				~	
Play	Pause 9	Step Stop	Dia	ignose			

Clicking Play button starts the script and all Statements executed are shown in Statements window below.

[sahi.ex	ample.com	m/_s_/spr/cont	roller8.	.htm		Sahi P	ro 9.7.0
	Set script	t from file	:					
	Script Dir	rectory:	C:/SahiPro/use	erdata/s	cripts/			~
	File:		sahi_sample_ru	ın.sah				
	🛃 Start I	URL:	http://sahitest.	com/de	mo/trair	ning/login.h	itm	
	Play	Pause	Step Stop	Dia	agnose			
	Statements: _navigateTo("http://sahitest.com/demo/training") _setValue(_textbox("user"), "test") _setPassword("password"), "*****") _click(_submit("Login")) _setValue(_textbox("q"), "1") _setValue(_textbox("q[1]"), "2") _setValue(_textbox("q[2]"), "2") _click(_button("Add")) _assertExists(_textbox("total")) _assertEqual("1400", _getValue(_textbox("total"))) _click(_button("Logout")) Stopped Playback: SUCCESS							
								ĥ

User can use Pause button to stop the script execution.

	Record Playback Clipboard Info Sahi Pro 9.7.
Sahi Training Site	Set script from file:
Username test	Script Directory: C:/SahiPro/userdata/scripts/
Password	Start URL: http://sahitest.com/demo/training/login.htm
Login	Play Pause Step Stop Diagnose
Not a user? <u>Register</u>	Statements: navigateTo("http://sahitest.com/demo/training")
	_setValue(_textbox("user"), "test")
Use test/secret to login	

Step button is used to run the script step by step.



Logging in Sahi

User can go through the logs using Logs link in Controller window.

View: Editor | Parsed Script | Browser Script | Logs

Logs contains suite runs, Average Time Taken By Browsers graph, Success Percentage By Different Browsers pie chart and all the Playbacks conducted.



User can go through the Sahi reports to know more about the execution process, the time taken for each action to run.

🗷 sahi_sample_run.sah x +	o x
🗧 🔶 🕐 🕐 kocalhost.9999/_s_/dyn/pro/DBReports_scriptReport?id=sahi_sample_run_chrome_948c2956058d70490f0b45a0bf0048a4bdd1 🖄 🖈	
Suites Suite Report Test Cases Report Script Report JS Code Coverage Report Suite Analysis	2
Script Name: sahi_sample_run.sah Auto Refresh 🗆	
Test Total Steps Failures Errors Success Rate Time Taken Node Load Browser Intermediate Statuses Final Status	
sahi_sample_run.sah 12 0 0 100 % 00:00:43 108 localhost:9999 0 Chrome 112.0.0.0 NA SUCCESS	
Report Id: bahi_sample_run_chrome497d10e107f18042e7090560bb7c53bc8981 Compare Logs Script_Info	
browserVersion=Chrome 112.0.0.;	
Starting script Expand All Collage All Show Falled 🗌 Show All Images 🖬 Convert To English 🗌	
_navigateTo("http://sahitest.com/demo/training") [1497 ms] [09:45:01.279 PM]	
_setValue(_textbox("user"), "test") [39939 ms] [09:45:41.218 PM]	
_set/sesword(_bassword(_bassword), ****) [141 ms][09:45:41.329 [M]]	
setValue(_textbox("q[1]"), "2") [111 ms][09:45:41.859 PM]	
_setValue(_textbox("q[2]"), "2") [125 ms] [09:45:41.984 PM]	
_click(_button("Add")) [231 ms] [09:45:42.215 PM]	
_assertExists(_textbox("total")) [124 ms] [09:45:42.339 PM]	
_assertVisible(_textbox("total")) [126 ms] [09:45:42.465 PM]	
_assertEqual("1400", _getValue(_textbox("total"))) [125 ms] [09:45:42:590 PM]	
_click(_button("Logout")) [233 ms] [09:45:42.823 PM]	
Stopping script	

Let's see a failure test case in the script. Changing quantity of book 3 in the cart will give us a different total. When the grand total is not equal to 1400, Sahi logs an error in the test case.

_setValue(_textbox("q[2]"), '10");

User can playback the script from Sahi Editor without reaching Controller. Select the script file, browser and start URL and click on Run button to start the playback.

hi_sample_run.sah c × 📫	Playback Properties		×
🏄 Auto Format 🛛 🞜 Create			
1 navigateTo("http://s	Script Directory:	C:/SahiPro/userdata/scripts/	
2 setValue(textbox("	Script File:	sahi_sample_run.sah	
3 _setPassword(_passv	Start Mode:	Browser 🗸	
4 _setValue(_textbox("	Browser:	chrome	
5 _setValue(_textbox("	Android Device:		
7click(button("Add"	Android Device.		
8 assertExists(textbo	IOS Device:	¥ ¥	
9 _assertVisible(_textb	Start URL:	http://sahitest.com/demo/training	
10 _assertEqual("1400",			
11 _click(_button("Logo			
12	Run in parallel The The Parallel The Para	hreads: 5	
	\odot Run sequentially in s	single browser session	
	O Bun distributed (mu	Itinla machines)	
	 Kull distributed (Ind 	iuple machines)	
	Show advanced settings		

Let's see a failure test case in the script. Changing quantity of book 3 in the cart will give us a different total. When the grand total is not equal to 1400, Sahi logs an error in the test case.

User can playback the script from Sahi Editor without reaching Controller.



Before execution of any step, Sahi waits for any Ajax activity, network activity to subside.

🜌 Books 🛛 🗙	+						U	×
\leftrightarrow \rightarrow C \land Not secure \mid secure	hitest.com/demo/trainir	ng/books.htm			\$	🔒 Inc	cognito	
	Title Core Java Ruby for Rails Python Cookbook	All avai	Attempt Attempt Cost Attempt Ks. 300 1 Ks. 200 2 Ks. 350 Tt Clear Logout	S diquantity to cart				
	Title Core Java Ruby for Rails Python Cookbook	M Quantity 1 2 10 Grand Total: 420	y Cart Unit Cost Rs.300 Rs.200 Rs.350	Total Cost Rs.300 Rs.400 Rs.3500				

If a step seems to be failing, Sahi waits for 2 seconds and tries to re-execute that step. It will do this 5 times, between the retries, if the system recovers, then Sahi will execute that step, else it'll mark the step as a failure.



Let's go through the logs, we see that script itself is in red.

De	Delete Compare Consolidate Report Hide Graphs Show Testcase Summary Configure columns Sync Filter Reset fil							Reset filters Do	vnload as 🗸	Download		
50	50 V H 4 Page 1 of 1 P H 0 Displaying 1 to 8 of 8 items											
					Choose V From	То 🕅	00:00:00 000		P F E Sk All	Choose 🗸		
	Script Name	Suite Name	Base Url	Browser	Start Time	End Time	Time Taken	Total Scripts	Status Summary	Status	Origin Machine	Comment
	» <u>sahi_sar</u>	nple_run.sah	http://sahit	chrome	Apr 25, 2023 11:57:32 AM	Apr 25, 2023 11:57:59 AM	00:00:26 120		Failed:1	FAILURE		0
	» <u>sahi_sar</u>	nple_run.sah	http://sahit	chrome	Apr 25, 2023 11:18:16 AM	Apr 25, 2023 11:18:26 AM	00:00:10 069		Passed:1	SUCCESS		0
	» <u>sahi_sar</u>	nple_run.sah	http://sahit	chrome	Apr 24, 2023 09:46:34 PM	Apr 24, 2023 09:47:15 PM	00:00:41 701		Aborted: 1	USER_ABORTED		0
	sahi_sar	nple_run.sah	http://sahit	chrome	Apr 24, 2023 09:44:59 PM	Apr 24, 2023 09:45:42 PM	00:00:43 135		Passed:1	SUCCESS		0

Opening the script, the assert step is marked in red and shows the expected and actual values mismatch. Opening onScritpFailure step shows the screenshot when the assert statement failure happened during test run.

z Editor - sahi sample_run.sah x z +	~ - o ×
🗧 🔶 C 🔘 localhost:9999/_s_/dyn/pro/DBReports_scriptReport2id=sahi_sample_run_chrome_e15dc27d0ce42047ab080c4031963504ba6f	
Sultes Suite Report Test Cases Report Script Report JS Code Coverage Report Suite Analysis	(
Script Name: sahi_sample_run.sah Auto Refresh 💈	
Test Total Steps Failures Errors Success Rate Time Taken Node Load Browser Intermediate Statuses Final Status	
sahi_sample_run.sah 16 1 0 94 % 00:00:21 318 localhost:9999 -1 Chrome 112.0.0.0 FAILURE FAILURE	
Report Id: sahi_sample_run_chrometcc901060688504ac80a542084/4d9cc/9ee Compare Logs <u>Script Info</u>	
Starting script Expand All Collapse All Show Failed 🗌 Show All Images 💋 Convert To English 🗌	
_navigateTo("http://sahitest.com/demo/training") [1038 ms] [11:57:35.151 AM] setValue(_texthox("user") "test") [303 ms] [11:57:35.454 AM]	
_setPassword(_password"), "*****") [246 ms][11:57:35.700 AM]	
_click(_submit("topin") [12 mg [11:57:35:832 AM] setValue (textbox("0", "1") [609 mg [11:57:35:832 AM]	
_setValue(_textbox("q[1]"), "2") [133 ms] [11:57:36.655 AM]	
_setValue(_textbox("q[2]"), "10") [127 ms] [11:57:36.782 AM] _click(_button("Add")) [229 ms] [11:57:37.011 AM]	
_assertExists(_textbox("total")) [124 ms][11:57:37.135 AM] assertWishlef textbox("total")) [126 ms][11:57:37.261 AM]	
_assertEqual(*1400*, _getValue(_textbox(*total*))) [13767 ms] [11:57:51.028 AM]	
[Assertion Failed] Expected:"1400"	
Actual: '4200' at: (/scripts/sahi_sample_run.sah&n=11)	
[+] onScriptFailure([+][object])	
_click(_button("Logout")) [79 ms] [11:57:55.337 AM] Stopping script	
[-] onScriptFailure([+][object])	
[] on CorintEsilura Dofnult/[1][abjact])	
_lockWindow() [158 ms] [11:57:51.186 AM]	
_focusWindow() [3176 ms] [11:57:54.362 AM]	
_takePageScreenShot() [833 ms] [11:57:55.195 AM]	
v teo sali sense rocuit x +	
K tota X + A © A hot ware ukłastowychodzkite A © 3	
· All available books	
Table In stock Cost Add quantity to cart Cose Issue 5 Rs. 100	
Rainy for Rain 12 Ba 200 2	
Coshbook 7 Ka.159	
My Cart	
Talla Quantity Unit Cost Tand Cost Cost Para 1 Re.100 Re.100	
Rany to Raini 7 Re.30 Re.300 Pythat Galacok I: 0 Re.350 Re.350	
Graal Total: (200	
Paylood S Totaas	
siUara	
shi ang	
🔢 🔍 Search 🔀 🗳 🖿 🕲 🔅 🤹 🌞 🖉 🖉 🦉 👰 🧕 transfer	

Library Functions

The steps that we see in the Sahi Script are low level instructions to the computer to perform actions on a browser.

The language of the business is much more in human speaking terms like – login to the system, add books quantities to the cart, check the total amount, and logout from the system.

So, we need to create Business Level Abstractions out of the steps that we recorded.

We create Business Level Abstractions using Functions in Sahi.

To create a function, we select steps that pertain to a logical business step and click on Create Function button, provide a function name and click on continue.

*sahi_sample_ru <u>n.sah</u> ¢	×
🔍 🛛 🎽 Auto Format 🛛 🎜 Cre	ate Function f Show Context
1 navigateTo("http:	Create Function /demo/training")
2setValue(textbo	x("user"), "test");
3 _setPassword(_pas	ssword("password"), "MgkKEQBU");
4 _click(_submit("Lo	gin"));
5 _setValue(_textbox	X("q"), "1"); x("q[1]"), "2");
Create Function	×
Function Name	
cartLogin	\sim
✓ Function Parameters (more	<u>e</u>)
✓ "test"	\$User
✓ "MgkKEQBU"	\$Password
🗆 Remove Duplicates 🗹 Crea	ate JsDoc
Add Function Call	
*sahi_sample_run.sah ¢ ×	
🔍 📝 Auto Format 🏾 🎜 Create Function 🏻 🎜	♦ Show Context
1 /** 2 * cartLogin(\$User, \$Password)	
3 * @param \$User - 4 * @param \$Password -	
5 */	ard)/
7 _setValue(_textbox("user"), \$l	Jser);
9click(_submit("Login"));	sword), \$Password);
10 } 11	
12 /*Functions Above */ 13	
14navigateTo("http://sahitest.com/	
LO GUILLOGIU COUL FIURILLOHU I.	/demo/training");
16	/demo/training");
16 17 _setValue(_textbox("q"), "1"); 18 _setValue(_textbox("q[1]"), "2");	/demo/training");
<pre>16 17 _setValue(_textbox("q"), "1"); 18 _setValue(_textbox("q[1]"), "2"); 19 _setValue(_textbox("q[2]"), "10"); 20 _click(_button("Add"));</pre>	/demo/training"););
<pre>16 17 _setValue(_textbox("q"), "1"); 18 _setValue(_textbox("q[1]"), "2"); 19 _setValue(_textbox("q[2]"), "10"); 20 _click(_button("Add")); 21 _assertExists(_textbox("total")); 22 _assertVisible(_textbox("total")); 23assertVisible(_textbox("total")); 24assertVisible(_textbox("total")); 25assertVisible(_textbox("total")); 26assertVisible(_textbox("total")); 27assertVisible(_textbox("total")); 28assertVisible(_textbox("total")); 29assertVisible(_textbox("total")); 20assertVisible(_textbox("total")); 20assertVisible(_textbox("total")); 21assertVisible(_textbox("total")); 22assertVisible(_textbox("total")); 23assertVisible(_textbox("total")); 24assertVisible(_textbox("total")); 25assertVisible(_textbox("total")); 26assertVisible(_textbox("total")); 27assertVisible(_textbox("total")); 28assertVisible(_textbox("total")); 29assertVisible(_textbox("total")); 20assertVisible(_textbox("total")); 20assertVisible(_textbox("total")); 21assertVisible(_textbox("total")); 22assertVisible(_textbox("total")); 23assertVisible(_textbox("total")); 24assertVisible(_textbox("total")); 25assertVisible(_textbox("total")); 26assertVisible(_textbox("total")); 27assertVisible(_textbox("total")); 28assertVisible(_textbox("total")); 29assertVisible(_textbox("total")); 20assertVisible(_textbox("total")); 20assertVisible(_textbox("tota</pre>	/demo/training"););
<pre>16 17 _setValue(_textbox("q"), "1"); 18 _setValue(_textbox("q[1]"), "2"); 19 _setValue(_textbox("q[2]"), "10"; 20 _click(_button("Add")); 21 _assertExists(_textbox("total")); 22 _assertVisible(_textbox("total")); 23 _assertEqual("1400", _getValue(_ 24 _etValue(_textbox("total"));</pre>	/demo/training");); _textbox("total")));
<pre>16 17 _setValue(_textbox("q"), "1"); 18 _setValue(_textbox("q[1]"), "2"); 19 _setValue(_textbox("q[2]"), "10"; 20 _click(_button("Add")); 21 _assertExists(_textbox("total")); 22 _assertVisible(_textbox("total")); 23 _assertEqual("1400", _getValue(_ 24 _click(_button("Logout")); 25</pre>	/demo/training");); _textbox("total")));

Sahi creates a function and extracts possible parameters from the recorded steps and takes them as arguments to the function.

In the recorded steps place, it will call the created function with the values from the steps.

We do similar procedure for remaining steps, combine the quantity adding steps to a single function.



_setValue(_textbox("q"), \$javaQ); _setValue(_textbox("q[1]"), \$rubyQ); _setValue(_textbox("q[2]"), \$pythonQ); _click(_button("Add")); } /* --Functions Above-- */ _navigateTo("http://sahitest.com/demo/training"); cartLogin("test", "MgkKEQBU"); addItemsToCart("1", "2", "10");



verifyGrandTotal("1400");
cartLogout();

To reuse the functions generated, we place them in a separate .sah script file.

Press Control and click on the function to open Context panel.



Context	\$ »
File Path:	
cart_library.sah 🗸	Open Include
Function Name:	
cartLogin	~
Arguments:	
\$User	
\$Password	
4	
Show JsDoc Insert 식	Insert Update
Accessor Repository (AR) Paramet	ers
File Path:	
~	Open Include
Key:	
	Edit AR Key
Value:	
	Save AR Value
Inline AR Key	

Select the library file from File Path where the function was defined and implemented. Click on Include button to add import statement to the current script.

1	_include("cart_library.sah");
2 3	<pre>navigateTo("http://sahitest.com/demo/training");</pre>
4	cartLogin("test", "MgkKEQBU");
5	
6	addItemsToCart("1", "2", "2");
/	verifyGrandTotal("1400").
9	
10	cartLogout();

Playback the script.

Let's look at the logs for the execution completed in Sahi Logs window.

Each function is listed in log and time for execution is listed after each step.

🜌 Editor - sahi, sample_run.sah x 🗷 sahi, sample_run.sah x +	~ - o ×
🗧 🔶 C 💿 localhost:9999/_5_/dyn/pro/D8Reports_scriptReport?id=sahi_sample_run_chrome_a405a00208b8d04abc0acd605cbd1183b544 🖄 🖈	
Suites Suite Report Test Cases Report Script Report JS Code Coverage Report Suite Analysis	😑 📧
Script Name: sahi_sample_run.sah Auto Refresh 🛛	
Test Total Steps Failures Errors Success Rate Time Taken Node Load Browser Intermediate Statuses Final Status sahi_sample_run.sah 12 0 0 100 % 00:00:04 127 localhost:s999 -1 Chrome 112.0.0.0 NA SUCCESS Report Id: sahi_sample_run_chrome_40fad06a0464204afb0beac0c5984b400674 Compare Logs Script Linfo Success Rate S	
Starting script Excend.All Collasse.All Show Falled Show All Images 2 Convert To English navigateTo("http://sahitest.com/demo/training") [1426 ms] [04:43:23.408 PM]	
_setValue(_textbox("user"), "test") [304 ms]] 4:43:22.712 PM] _setPassword(_password"), ******) _click(_submit("togin")) [157 ms] [04:43:24.00 PM]	
[+] addItemsToCart("1", "2", "2")	
[+] verifyGrandTotal("1400")	
(+) cartogot()	
ցուրին էր չու լիչ	

Let's change the script a little to raise an error in grand total assertion, and see the logs.

verifyGrandTotal("1450)");							
Surprivatile Surre Name Dase of Drowser	Julit Time	Litu nine	Time laken	Scripts	Status Summary	Jidius	Machine	Comment
» sahi_sample_run.sah http://sahit chrome	Apr 25, 2023 04:49:13 PM	Apr 25, 2023 04:49:37 PM	00:00:24 102		Failed:1	FAILURE		0
Z Editor - sahi_sample_run.sah 🗙 Z sahi_sample_run.sah	× +						~ -	o ×
\leftarrow \rightarrow C () localhost:9999/_s_/dyn/pro/DBReports_script						☆ 😐 👽	ë 40 🖈	🛛 🎲 E
Suites Suite Report Test Cases Report Script Report	JS Code Coverage Report Su	ite Analysis						2
Script Name: sahi_sample_run.sah Auto Refresh 🗹								
Test Total Steps Failures Errors Success	s Rate Time Taken	Node Load Brow	vser Intermedia	te Statuses Final Status]			
sahi_sample_run.sah 16 1 0	94 % 00:00:16 862 localho	st:9999 -1 Chrome 112.0	.0.0	FAILURE FAILURE	1			
Report Id: sahi_sample_run_chromec963aa3106b1004f9	8086ed0cb6ffe799e71 Comp	bare Logs						
					, ,			
Starting script Expand All Collapse All Show Failed Sh	ow All Images 🗹 Convert To	English 🗌						
_navigateTo("http://sahitest.com/demo/training") [1268	8 ms] [04:49:16.209 PM]							
[+] cartLogin("test", "MgkKEQBU")								
[+] addItemsToCart("1", "2", "2")								
[+] verifyGrandTotal("1450")								
[+] cartLogout()								
Stopping script								

Grouping as functions lets us detect the code where the assertion failed. Screenshot is also provided in logs for the failure occurred.



Scenarios in Sahi

Sahi Scripts are written in JavaScript language.

Testers are present where they understand the business functionality of the application, but do not necessarily understand the syntax and semantics of Sahi Script.

To help them participate in the automation process, we have an alternative way of defining our test cases and scenarios. We do that using a scenario file.

Click on New button and select Scenario button.



Click on Save button and save the file.

sahi_s	sahi_sample_run.sah & * *new 1.s.csv & *									
🔍 Ins	Insert testcase Load Sahi Library Function Details Import Insert Tags Column Insert Comment Column									
	А	В	С	D	E	F	G	Н	Ι	
1	TestCase	Key Word	Argument 1	Argument 2						
2										
3	My First		My testcase							
	Testcase	[Documentation]	description.							
			Modify as							
			needed							
4		My Function	my argument	my argument 2						
5										
6										
7										
-										

Save As		
Script Directory:	C:/SahiPro/userdata/scripts/	~
File:	cart	
Save Ca	ancel	

Give a name in the test case column.

	А	В	С	D	E
	TestCase	Key Word	Argument 1	Argument 2	
:					
	Test Grand Tota	Documentation]	My testcase description. Modify as needed		
;					

А	В	С	D	E	
TestCase	Key Word	Argument 1	Argument 2		
Test Grand		Add books to ca			
Total	otal [Documentation]				

Add a description to the test case in Argument 1 column.

User can use different functions available in cart_library file by including in the current file.

_

А	В	С	D	Е
TestCase	Key Word	Argument 1	Argument 2	
Test Grand		Add books to		
Total	[Documentation]	cart and verify		
_		total		
	cart			
	cart Login			
	cart Logout			
	add Items To			
	v.			

Ctrl+click on the function to know more details about it

Function Details		E X
File:		
cart_library.sah		~
		Open Include
Function Name:	cartLogin	
Arguments:		
\$User		
\$Password		
✓ Set as field:value		OK Cancel

Select Include button to include cart_library.sah in the current scenario file.

Now user can enter the data that is needed for login to happen i.e., username and password in Argument 1 and Argument 2 columns.

3	Test Grand Total	[Documentation]		Add books to cart and verify total			
4		cartLogin	V	test		MgkKEQBU	
5			V				

User can pass the parameters values in Function Details window.

	А	В	С	D	Е	F
1	TestCase	Key Word	Argument 1	Argument 2		
2		loadSahi 🛛 🔍	cart_library.sah			
3		V				
4	Test Grand		Add books to			
	Total	[Documentation]	cart and verify			
			total			
5		cartLogin 🛛 🔻	test	MgkKEQBU		
6			<i>java Q</i> : 1	<i>ruby</i> Q : 2	python Q:2	
		addItemsToCart				
7		verify Grand 🔍	<i>Total</i> : 1400			
		Total	_			
8		cart Logout 🛛 🔻				
9		V				

Let's run the script.



Opening logs

Editor - cart.s.csv X	Z cart.s.csv	× +				~ - o x
$\leftarrow \rightarrow$ C () localhost:9999/_					№ ☆	
Suites Suite Report Test Cases	Report Script Report JS Cod	e Coverage Report Suite Analys	sis			2
Script Name: cart.s.csv Auto F	Refresh 🔽					
Test Total Steps Failures E	rrors Success Rate Time Tak	en Node Load	Browser Intermediate Statu	ises Final Status		
cart.s.csv 11 0	0 100 % 00:00:04 1	71 localhost:9999 -1 Chror	me 112.0.0.0	NA SUCCESS		
Report Id: cart_chrome4125f13 Script_Info	10e9f004d79091c4021f4425bc4	81 Compare Logs				
The Test Course Descend Follow	Current Batta					
1 1 0	1 Success Rate					
Test Case Id	Description Time Taken	Intermediate F Statuses Sta	inal atus			
Test Grand Add books t	o cart and verify 00:00:02 total 748	NA SUCCE	ESS			
			-			
Starting script Expand All Collaps	e All Show Failed 🗌 Show All	Images 🗹 Convert To English [
	loadSahi	cart_library.sah				
Test Grand Total	[Documentation]	Add books to cart and				
		veniry total				
	cartLogin	test	MgkKEQBU			
	addItemsToCart	java Q:1	ruby Q:2	python Q:2		
	verify Grand Total	Total:1400				
	cart Logout					
Stopping script					•	

Clicking the step name shows the function called and the code executed.

	cartLogin	test	MgkKEQBU
[-] cartLogin("test", "MgkKE	QBU")		
_setValue(_textbox("user"), "test")	1553 ms] [05:14:31.951 PM]	
_setPassword(_password('	'password"), '	"****") [239 ms] [05:14:32.190 PM]	
_click(_submit("Login"))	[205 ms] [0	5:14:32.395 PM]	

Data Driven Suites in Sahi

Let's run our scripts in batches using Sahi Parallel playback.

To execute multiple scripts in a batch, we need to create a suite file.

We create a suite file by first choosing the scripts that we want to execute.

Now click on Create Suite \rightarrow Data Driven Suite

Editor - cart.s.csv	×	🖅 cart.	S.CSV		×	+
$\leftarrow \rightarrow \mathbf{C}$ (i) localhost:9999)/_s_	/spr/edit	or/editor.h	tml?dir=C%	3 A%2 F	SahiPro%
🎦 New 🔻 🖺 Open II Save 🔻	(Create S	uite 3	Create AR	👬 Fin	d Usage
Navigator 🕀 🖻 🛱 🤹		Suite		ah c× c	art.s.cs	sv c x
4	DD CSV	Data Dr	riven Suite	Load Sa	hi Librar	y Func
scripts (11)			А	E	3	С
Image: Ample and Ample		1	TestCase	Key Wo	rd	Argume
apis.sah		2		loadSah	i v	cart_libra
Cart.s.csv 1		3				
<pre>cart_library.sah cart_recording.sah global_include.sah login.sah cart_library.sah</pre>		4	Test Grand Total	[Docum	▼ entation]	Add boo cart and total
new_recording.sah		5		cartLogi	n v	test
📄 🗔 polymer-apis.sah 📄 🔲 sahi_sample_run.sah		6		addItem	▼ sToCart	java Q∶
		7		verify Gi Total	rand 🔍	<i>Total</i> : 1
		8		cart Log	out	

The above step creates a suite file. Save the file.

Generation G			cov o cur	L_IIDrary.san 6 #	*new 2.dd	CSV C X			
		_							
▶ □ demo (264) 1 ▶ □ java (3) 2 ■ apis.sah 2 ■ ✓ cart.s.csv 3 ■ □ first_recording.sah 4 ■ □ first_recording.sah 5	А	В	С	D	Е	F	G	Н	I
Image: provide state Image: provide state	#script	url	tags	depends	prerequisites	timeout_minutes			
Cart.s.csv Cart.sibrary.sah Cart_library.sah Cart_librar	cart.s.csv								
Cart_library.sah	login.sah								
first_recording.sah Jobal_include.sah									
login.sah 6									
new_recording.sah 7									
polymer-apis.sah 8									
sahi_sample_run.sah									
10									
11									
12									

Let's playback the suite using 'Playback' button.



Playback happens parallelly opening 2 edge browsers.

Playback Status	×				
Testcases: 1/1, Scripts: 2/2 BROWSER: edgenew					
cart_test_suite.dd.csv					
SUCCESS on edgenew					
cart_test_suite.dd.csv					
	-				

Logs show the graph showing the scripts and their run time. User can see the time taken for running the scripts. Clicking on the test link, Sahi opens detailed report of the execution.

z cart_test_suite.dd.cs	v ×	ŀ														
\leftrightarrow \rightarrow \odot \odot k	calhost:9999/_s_/dy													e 🔹 🖈 🛛	• 🭕	: 6
Suites Suite Report Tes	t Cases Report JS Cod	e Coverage	e Report	t Suite Analysis											-	
107												[
Run Time (sec)							Sector									
							Scripts									
Suite Name: cart_test_su	ite.dd.csv Auto Refr	esh 🛃														
Browser Type edgenew Start Time Apr 25, 2023 End Time Apr 25, 2023 Time Taken 00:00:14 05	3 05:34:34 PM 3 05:34:48 PM 0	Total scr Scripts p Scripts f	ipts passed /ailed errored	2 2 0												
Suite Info		Scripts a Scripts a Scripts a Scripts r	ikipped aborted aborted not sup;	0 0 timeout 0 ported 0												
Nodes Info		Status		SUCCESS												
Scripts:	Scripts:															
Compare Screenshots Co	ompare Logs Expand	All Col	lapse Al	JI										Download as 🛩	Downl	load
Test	Start Url	Steps: Total/Fail	lures/Er	mors/%	Testca Total/	ases: Passed	/Failed/%	Start Time	Time Taken	Node	Load	Intermediate Statuses Healed, Failure, Error, NA	Final State	IS ilure,Error,Abort 🗸		
▼ ■ cart_test_suite.dd.csv	/													_		
Cart.s.csv	http://sahite			100 %			100 %	Apr 25, 2023 05:34:35 PM	00:00:05 771	localhost:9999			SUCCESS			
login.sah	http://sahite			100 %				Apr 25, 2023 05:34:38 PM	00:00:05 288	localhost:9999			SUCCESS			

Data Driven Suites Logs Detailed View

Suites Suite Report Test Case	es Report Script Report JS Co	ode Coverage Report Suite	Analysis		
Script Name: cart.s.csv Aut	o Refresh 🔽				
Test Total Steps Failures	Errors Success Rate Time	aken Node Load	Browser Inte	In Success	
Report Id: cart test suite edge	enew 375218090cedb0414d095	3001e9b553145bc Compar	re Logs		
Script Info					
Total Test Cases Passed Fai	led Success Bate				
1 1	0 100 %				
Test Case Id	Description Time Take	n Intermediate Statuses	Final Status		
Test Grand Add book	s to cart and verify 00:00:0	2 NA	SUCCESS		
Iotal	total 60				
Starting script Expand All Colle	apse All Show Failed 🗌 Show A	II Images 🗹 Convert To En	glish 🗋		
	loadSahi	cart_library.sah			
Test Grand Total	[Documentation]	Add books to cart and verify total			
		veniy total			
	cartLogin	test	MgkKEQBU		
[-] cartLogin("test"_"Mok	KEOBU")				
sot)/alug/_taythay/"us	or") "teet") [2241 me][05:24	IMA 656 051			
_setPassword(_passwor	rd("password"), "*****") [194	ms] [05:34:39.416 PM]			
_click(_submit("Login"))) [297 ms] [05:34:39.713 PM]				
	addItemsToCart	java Q:1	ruby Q:2	python Q:2	
	verify Grand Total	Total: 1400			
	cart Logout				
Stopping script					-
Suites Suite Report Tes	st Cases Report Script Re	port JS Code Covera	ge Report Suite Anal	ysis	
Script Name: login.sah	Auto Refresh 🗹				
	J				
Test Total Steps Fa	ailures Errors Success Rat	e Time Taken	Node Load	Browser Intermediate S	tatuses Final Status
login.sah 5	0 0 100 9	6 00:00:05 288 local	host:9999 -1 Edge	e 112.0.1722.58	NA SUCCESS
Report Id: cart_test_suite	e_edgenew375218090ce	db0414d0953001e9b55	53145bc Compare Lo	gs	
<u>Script Info</u>					
Starting script Expand A	All <u>Collapse All</u> Show Failed	🗌 Show All Images 🕻	Convert To English		
_navigateTo("http://sahi	test.com/demo/training")	[3793 ms] [05:34:41	.822 PM]		
_setValue(_textbox("use	r"), "test") [383 ms] [0	5:34:42.205 PM]			
_setPassword(_password	d("password"), "*****")	[133 ms] [05:34:42.33	38 PM]		
_click(_submit("Login"))	[220 ms] [05:34:42.55	8 PM]			
_click(_button("Logout")) [723 ms] [05:34:43.2	at hul			
Stopping Script					

BDTA – Business-Driven Test Automation

Business-Driven Test Automation (BDTA) allows test automation to begin much earlier in the project lifecycle - right at the feature conceptualization stage. The application under test need not be ready to begin BDTA. With BDTA, prior to the readiness of your feature or application, we can:

1. Specify the desired application flow using straightforward language, employing key words for various actions (e.g., login, add books, etc.).

- 2. Incorporate optional parameters into different steps based on specific testing requirements, allowing for future adjustments or additions when the application is fully developed (e.g., login | username: test | password: |).
- 3. Implement data-driven testing for scenarios that involve extensive data manipulation, enabling the creation and population of relevant data input files.
- 4. Organize our tests and scenarios into suites, ensuring a structured and systematic approach to test automation management.
- 5. Utilize tags to logically separate tests and enhance control over their execution, enabling better test management and organization.
- 6. Seek verification and validation of our test automation efforts by involving peers and managers to review and provide feedback.
- 7. Version control your test automation artifacts by checking them into a designated repository, ensuring proper tracking and management of changes over time.

BDTA Implementation using Sahi:

let us imagine that we are going to work on a simple shopping application wherein the user can purchase books and the use cases to add books to their cart and verify the cart total amount now let us see how we can implement it using business driven test automation.

Open Sahi Pro Editor. We first create a new scenario file. Our first use case is login we write that in Key Word column, a login will need a username and a password typically so we can add that data as a parameter name:value in Argument 1 & argument 2 columns.Next step is to add books, we do not yet know what books we will be adding so we can leave parameters empty.

А	В	С	D	E
TestCase	Key Word	Argument 1	Argument 2	
TC01		Verify cart total		
Shopping Cart	[Documentation]	after adding		
		books		
	User Login 🛛 🔻	username :	password :	
		test		
	Add 🔹			
	Programming			
	Books			
	Verify Total 🔍			
	Amount			
	User Logout 🔍			

Save the file.

Script Directo	C:/SahiPro/userdata/scripts/	~
File:	ShoppingCart	

Now we will begin implementing the key words listed in the above file.

Right click on the test case name and select 'Run/Record'. Enter start URL and click Run/Record.

Script Directory:	C:/SahiPro/userdata/scripts/	
Script File:	ShoppingCart.s.csv	
Test Case 💙	TC01 Shopping Cart	~
Start Mode:	Browser	~
Browser:	chrome	~
Android Device:		× 🕏
iOS Device:		× 🕏
Start URL:	http://sahitest.com/demo/training	~

Now about run record this feature runs the selective test case with the intent of recording any missing or non-implemented keywords or functions as you begin it looks in the entire database of our Sahi scripts looks for a function with the keyword name and if not found prompts you to record it by opening the Sahi pro controller.

🜌 Login 🛛 🗙 🕂		🗸 — 🗆 🌌 Sahi Controller - Google Chrome — 🗆 🗙
\leftarrow \rightarrow α A Not segme L subject on	Mama Itraining I agin htm	★ □ → Incognite (2) ★ sahi.example.com/_s_/spr/controller8.htm
A Not secure samtes.com	ng demoy a ainin ghogi karan	k k k k k k k k k k k k k k k k k k k
	Sahi Training Site	Script Directory: C:/SahPro/userdata/scripts/ V Script Name: ShoopingCartScriptsah C AR File: ShoopingCartScriptsah C Keyword: User Login
	Password Login Not a user? <u>Register</u>	Press CTRL key and hover mouse over any element. Accessor: •• div("Sahi Training Site Invalid username or) Alternatives: v Value: Sahi Training Site Invalid username or password t
	Use test/secret to login	Prefix: Hover Mode: BROWSER Session key: session1d Assert Ust Properties Other Actions: Choose
		Evaluate expression Recorded steps Diagnose Dacs Editor

After login is done, select red button to stop recording.

					Record Createrni Playback Clipboard Into Can	
					Recording User Login in ShoppingCartScript.sah	•
	All av	ailable bo	oks		Press CTRL key and hover mouse over any element. Accessor:	g : User Logir toppingCar top recordi
Title	In stock	Cost	Add quantity to cart		Alternatives:	Hilight
Core Java	5	Rs. 300	0		Value: Sahi Training Site Invalid username or password	Set
Ruby for Rails	12	Rs. 200	0		Prefix:	Hover
Python	7	D- 250			Mode: BROWSER Session key: sessionId	
Cookbook	/	Ks. 350	0		Assert List Properties 🔂 Other Actions: Choose	~
	Add	Clear	ut		APIs List:	
					Evaluate expression Recorded steps Diagnose D	ocs Edito
					_click(_submit("Login"));	-
	I	My Cart				
Title	Quantity	Unit Cos	t Total Cost			
	Grand Total:	0				
					Sahi-Browser ♥ → Append HighlightChoose ♥	Histor

🌌 Sahi Con	troller - Google	Chrome			—		×
🔒 sahi.e:	xample.com/_	_s_/spr/cont	roller8.htm				
Record	Create Fn	Playback	Clipboard	Info	Sa	hi Pro	9.7.0
Function /** * UserLo * @para	i: ogin im \$username	-					•
function setV click }	UserLogin(\$us 'alue(_textbox('assword(_pass :(_submit("Log	ername) { "user"), \$use word("passw in"));	ername); /ord"), "MgkKE	EQBU");			
☐ Funct ✔ \$use ☐ \$Pas	tion Paramete ername ssword	rs (<u>more</u>) "te "M	est" IgkKEQBU"				
Pass Save	oy object (Re-record	Remove [Abort	Duplicates				

Sahi opens Create Function tab to tell us the parameters used. We can select the additional parameters if the required attribute is not default selected. The default selection is done when the parameter value in Sahi scenario file matches the input value used by the user.

Click on save to save the function.

Sahi Controller - Google Chrome	—		×
sahi.example.com/_s_/spr/controller8.htm			
sahi.example.com says Recording Keyword: Add Programming Books Script: ShoppingCartScript.sah To proceed, click OK. To modify, click Cancel and enter details as required. OK Assert List Properties To Other Actions: Cho	cose	Cance	el
APIs List:			
Evaluate expression Recorded steps Diac	<u>jnose</u>	Docs	<u>Editor</u>

Now if the next step in the scenario is already implemented, Sahi continues the playback, else it will prompt to record the function.

sabitast.com/dama/training/books.htm	↔ 🗖 🚗 Incognito (2)	saniexample.com/_s_spi/conditiero.num
Surresteen) deno, daning, books.nan		Record Create Fn Playback Clipboard Info Sahi Pro 9.7.0
All available books Title In stock Cost Add quantity to cart Core Java 5 Rs. 300 1 1 Ruby for Rails 12 Rs. 200 2 1 1 Python 7 Rs. 350 3 1 1 Cookbook 6 Add Clear Logout		Record Create Fn Playback Clipboard Info San Pro 9.7.70 Recording Add Programming Books in So So So So Press CTRL key and hover mouse over any element. Accessor: Info Click Alternatives: Click Alternatives: Info San Pro 9.7.70 Click Hilight Value: Sahi Training Site Invalid username or password I Set Prefix: BROWSER Session key: sessionId Assert List Properties Other Actions: Incose Inc
		Evaluate expression Recorded steps Diagnose Docs Editor
My Cart		_click(_button("Add"));
Title Quantity Unit Cost Total Cost Core Java 1 Rs.300 Rs.300 Ruby for Rails 1 Rs.200 Rs.200 Purbor Coekbook 1 Re 250 Rc 250		
Grand Total: 850		Sahi-Browser V - Append HighlightChoose V History Appended: _assertKists(_textbox("total")); _assertKight(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));

Now click on stop record button to see the recorded steps in the function.

Record Create Fn Playba	ck Clipboard	Info	Sahi Pro 9.7.0
Function: /**			
* AddProgrammingBooks	-0.6		
_click(_row("Core Java5Rs. : _setValue(_textbox("q"), "1" _click(_row("Ruby for Rails1 _setValue(_textbox("q[1]"), _click(_row("Python Cookboo _setValue(_textbox("q[2]"), _click(_div("All available boo _click(_button("Add")); }	9() % 300")); 2Rs. 200")); "1"); ok7Rs. 350")); "1"); ks TitleIn stockCost	tAdd quar	ntity to cart Cc
4			•
Function Parameters (more	2)		
1 \$Q	"1"		
2] \$Q1	"1"		
<mark>3</mark> □ \$Q2	"1"		
🗌 Pass by object 🗌 Remo	ve Duplicates		
4 Save Re-record Abort			

Now Sahi will prompt us to record Verify Total Amount function.

sahi.example.com/_s_/spr/controller8.htm									
sahi.example.com says Recording Keyword: Verify Total Amount Script: ShoppingCartScript.sah To proceed, click OK. To modify, click Cancel and enter details as required.									
OK Cancel									
Assert List Properties D Other Actions: Choose V									
APIs List:									
Evaluate expression Recorded steps Diagnose Docs Editor									
_click(_button("Add"));									
Sahi-Browser \checkmark \rightarrow Append HighlightChoose \checkmark \Box <u>History</u>									
Appended: _assertExists(_textbox("total")); _assertVisible(_textbox("total")); _assertEqual("1400", _getValue(_textbox("total")));									

Hover on total amount element

sani.e	xampie.	.com/_	s_/spr/cor	itroller8.ntm			
Record	Create	e Fn	Playback	Clipboard	Info	Sah	i Pro 9.7.0
Recordin	g Verify	/ Total	Amount in	ShoppingCa	artScript.	.sah	• •
Press CT	RL key	and ho	ver mouse	over any ele	ement.		
Accesso	r: 🔶 🗰 🚊	textbo	k("total")			ů	Click
Alternat	ives:	_textbo	x("total")			~	Hilight
Value:	8	350					Set
Prefix:							Hover
Mode:	В	BROWSE	ER	Session key	/: se:	ssionId	
Assert	List Pr	ropertie	s 🛃 (Other Actions	s: Cho	ose	~
Evaluate	e expre	ession dd"));	Recorde	d steps Hiablight	Diag	<u>se</u> ►	Docs Editor
Appended _assertExi _assertVis _assertEqu	sts(_tex ible(_tex ual("140	tbox("t xtbox(" 0", _ge	otal")); total")); tValue(_te;	ktbox("total")));		

And add assert statements.

Evaluate expression	Recorded steps		<u>Diagnose</u>	<u>Docs</u>	<u>Editor</u>
_assertExists(_textbox("t _assertVisible(_textbox(" _assertEqual("850", _get	otal")); total")); Value(_textbox("tota	al")));			^
Sahi-Browser ♥ →	Append Highlig	ht	Choose 🗸		<u>listory</u>
Appends the co	ntents of 'Evalute exp	pression	' textbox to t	he scrip	t being re

Append the assertions using Append button to the current function. Stop the recording.

sahi.example.com/_s_/spr/controller8.htm									
Record	Create Fn	Playback	Clipboard	Info	Sahi Pro 9.7.0				
Function /** * Verify */ function 1 _asse _asse }	TotalAmount VerifyTotalAmo ertExists(_textb ertVisible(_textbertVisible(_textbertEqual("850",	punt() { pox("total")); box("total")); _getValue(_	; textbox("total	")));					
2 Funct \$Tot	tion Paramete al	rs (<u>more</u>) "8] Remove [50" Duplicates						
Save	Re-record	Abort							

Add Total to function parameters and save the function.

Sahi will prompt us to record Logout functionality in User Logout function.



Stop the recording and save the function.

This will close the Sahi controller and goes back to editor.

Shop	pingCart.s.csv	c ×												•
🔍 In	sert testcase	Load Sahi Librar	y Function Details	Import Inse	rt Tags Column	Insert Comr	nent Column					Sample scen	ario 👔	i
	А	В	С	D	E	F	G	н	I	J	К	L	м	^
1	TestCase	Key Word	Argument 1	Argument 2										
2		loadSahi	ShoppingCartScript.sah											
3		Ŧ												
4	TC01 Shopping Cart	[Documentation]	Verify cart total after adding books											
5		User Login	username : test	Password : MgkKEQBU										
6		Add Programming Books	Q : 1	Q1 : 1	Q2 : 1									
7		Verify Total Amount	Total : 850											
8		User Logout												
9														
10														
11		v												
12														
13														
14														
15														
16										Dlauback	Ctatue			
17										PIdyDack	Status			<u> </u>
18										Testcase	s: 0/0, Scripts:	0/1 BROWSER:	chrome	
19 20										Shopping	JCart.s.csv			
21										SUCCES	S on chrome			Ĩ.
22										Shopping	Cartic cov			
222							1			Shopping	JCart.S.CSV			

We can ctrl+click on ShoppingCartScript.sah to see the script file and implemented functions.



User can now run the playback using Playback functionality.

Script Directory:	C:/SahiPro/us	erdata/scripts/		
Script File:	ShoppingCart.	.s.csv		
Test Case 💙 :	TC01 Shoppin	ıg Cart		~
Start Mode:	Browser			~
Browser:	chrome			~
Android Device:				✓
iOS Device:				✓ 参
Start URL:	http://sahites	t.com/demo/trainin	g	~
 Run in parallel Run sequential 	Threads: 5	ession		
\bigcirc Run distributed	(multiple machines)			
Reset 🖁 🛣 Crea	e Ant Target			Run •
Reset * Crea	e Ant Target + st.com/demo/training/books.ht	m available books		Run
Reset Crea	e Ant Target + st.com/demo/training/books.ht All	m available books		▶ Run
C A Not secure sahit	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5	m available books Cost Add qu Rs. 300 1	nantity to cart	
Reset Crea	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5 Ruby for Rails 12 Puthon	m available books Cost Rs. 300 Rs. 200	lantity to cart	Run Run T
Reset Crea	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5 Ruby for Rails 12 Python 7 Cookbook 7	The second state of the se	iantity to cart	
Reset * Crea	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5 Ruby for Rails 12 Python 7 Cookbook 7 Ad	available books Cost Add qu Rs. 300 1 Rs. 200 1 Id Clear Logout My Cart	antity to cart	
Reset Crea	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5 Ruby for Rails 12 Python 7 Cookbook 7 Inte Quantity Core Java 4	m available books Cost Rs. 300 Rs. 200 Rs. 200 1 1 1 1 1 1 1 1 1 1 1 1 1	Total Cost	
Reset * Crea	e Ant Target + st.com/demo/training/books.ht All Title In stock Core Java 5 Ruby for Rails 12 Python Cookbook 7 Title Quantity Core Java 1 Ruby for Rails 1 Python Cookbook 1	available books Cost Rs. 300 Rs. 200 Rs. 200 Clear Logout Clear Logout My Cart Unit Cost Rs.300 Rs.200 Rs.300 Rs.200 Rs.350	Total Cost Rs.300 Rs.200 Rs.350	



Seeing logs will report all the steps executed.

$\leftarrow \rightarrow \mathbf{C}$ (1) localhost:9999	/_s_/dyn/pro/DBReports_scriptRe	eport?id=ShoppingCart_chrome_9	03e703607cde	e041110b9810708	3580777ab0		É
Suites Suite Report Test Case	s Report Script Report JS	Code Coverage Report Suite /	Analysis				
Script Name: ShoppingCart.s.	csv Auto Refresh 🗌						
Test Total Steps	Failures Errors Success Ra	te Time Taken Noo	de Load	Browser	Intermediate Statuses	Final Status	
ShoppingCart.s.csv 15	0 0 100	% 00:00:04 842 localhost:999	99 -1 Chr	ome 112.0.0.0	NA	SUCCESS	
Report Id: ShoppingCart_chrom Script Info	ea89c7e6609c2904a8a098	3d0548b23f204ad Compare Lo)gs				
Total Test Cases Passed Fail	ed Success Rate						
Test Case Id	Description Time T	aken Intermediate Statuses	Final Status				
TC01 Shopping Verify ca	art total after adding 00:0	10:03 NA	SUCCESS				
Cart	DOOKS	396	لا يوروسورو				
Starting script Expand All Colla	I <u>pse All</u> Show Failed 🗌 Shov	v All Images 🗹 Convert To Eng	glish 🗌				
	loadSahi	ShoppingCartScript.sah					
TC01 Shopping Cart	[Documentation]	Verify cart total after adding books					
	User Login	username:test	Passv	vord:MgkKEQBL	J		
	Add Programming Books	Q:1	Q1:1		Q2:1		
	Verify Total Amount	Total:850					
	User Logout						
Stopping script	· · · · · · · · · · · · · · · · · · ·						
		-					
	User Login	username:test		Password:Mg	gkKEQBU		
[-] UserLogin("test", "M	lgkKEQBU")						
_setValue(_textbox("u	user"), "test") [1506 ms]	[06:43:49.734 PM]					
_setPassword(_passw _click(_submit("Login	ord("password"), "*****") "))	[197 ms] [06:43:49.931 Pl 176 PM]					

Data Drive Test Cases in Sahi

We take the scenario we implemented further by data driving our tests. Testers often need to keep the data separate from their test cases, there could be two reasons, firstly we may need to access the same data for multiple test cases and secondly a test case or some of its steps

may need to be repeated with multiple sets of data. So, we need a framework to enable a test case to be run repeatedly with multiple sets of data. Let's data drive the login scenario using different data.

Right click on the key word and select Data Drive.

Cart	[Documentation of the second s	on]	adding bo	onj adding books						
	User Login	V.	usernam	e : test	Passwo	ord :				
		Ins	ert row abo	ve		ΣΒΠ				
	Add	Ins	ert row belo	w						
	Programmi Books	Insert column left								
	Verify Total	Ins	ert column	right						
	Amount	Remove row								
	User Logoi	Remove column								
		Un	do							
		Re	do							
		Ru	n / Record							
		Da	ta Drive							
		Da								

We'll get a pop up to enter data file name and label.

TU.			
)B Data	Drive		×
Dat	ta File 🗌 :	· · · · ·	
Lat	bel:	~	
Dat	ta:		
	Column Header	Value	
_	username	+ test	
	Password	+ MgkKEQBU	
_			
		V	Save

Label: Credentials Data: Column Header Value Username + test	Dat	a File 🗌 :	ShoppingCartData	~	· 🕒	
Data: Column Header Value Username + test	Lab	el:	Credentials		\sim	·
 Column Header Value username test 	Dat	a:				
 ✓ username + test ✓ Decompared + Malu/SOBU 	✓	Column H	leader		Value	
	✓	usernam	e	test		
Password + MigkkeQBU	✓	Password	1	+	MgkKEQBU	
						🧹 Sa

Click Save.

TC01			Verify cart total after		
Shopping Cart		[Documentation]	adding books		
	[ShoppingCartData.csv:Credentials]	User Login 🔍	username : test	Password :	
		J		MgkKEQBU	
		Add	Q :1	Q1 :1	Q2 :1
		Programming			

Let's see data drive for multiple rows. Selecting Add Programming Books and Verify Total Amount steps. Right click and select data drive.

Shopping Cart		[Documentai	ionj	adding books					
	[ShoppingCartData.csv:Credentials]	als] User Login 🔻 use		username : test Passwo MgkKE0		vord : EQBU		,	
		Add Programmin Rooks	~ ∼	Q : 1 rt row above	Q1 : 1		Q2 : 1		
		Verify Tot Amount	Insei Insei	rt row below rt column left					
		User Log	Inse	rt column right		-			
			Rem Rem	iove rows iove columns					
			Undo Redu	0					
			Run	/ Record					
			Data	Drive					
									Play

)ata I	Drive			×
Dat Lab	a File : ShoppingCartData el: BooksQuantities			
Dat	a:			
<	Column Header	Value		
<	Q +] 1		
~	Q1 +] 1		
✓	Q2 +] 1		
~	Total +	850		
			🖌 Sav	'e

Click Save button.

Now we can see that both steps are wrapped in [Repeat] [End] loop.

[ShoppingCartData.csv:Credentials]	User Login 🔍	username : test	Password : MgkKEQBU		
[ShoppingCartData.csv:BooksQuantities]	[Repeat]				
	Add	Q :1	Q1 : 1	Q2 : 1	
	Programming				
	Books				
	Verify Total	Total: 850			
	Amount				
	[End]				
	User Logout				

ShoppingCartData file can be seen by clicking ctrl+file name in the current file.

Shopp	ShoppingCart.s.csv c × ShoppingCartData.csv c ×									
Q										
r	А]	В	С	D	E			
1	Credentials		userna	me	Password					
2			test]	MgkKEQBU					
3										
4	BooksQuantiti	es	Q		Q1	Q2	Total			
5			1		1	1	850			
6										
7										

Now let's add some more data to BooksQuantities to run the test multiple times.

	А	В	С	D	Е	F
1	Credentials	username	Password			
2		test	MgkKEQBU			
3						
4	BooksQuantities	Q	Q1	Q2	Total	
5		1	1	1	850	
6		10	5	3	5050	
7		0	0	0	0	
8		-2	ab	-5	0	
9						
10						

Let's run the test case. Right click on the test case name and select Run/record.

TC0	1 Insert row above	
Sho	Insert row below	ntial
	Insert column left	nuar
	Insert column right	Qua
	Remove row	
	Remove column	
	Undo	
	Redo	
	Run / Record	
)	Data Drive	

Playback Properties			×
Script Directory:	C:/SahiPro/userdata/script	s/	
Script File:	ShoppingCart.s.csv		
Test Case 💙 :	TC01 Shopping Cart		~
Start Mode:	Browser		~
Browser:	chrome		~
Android Device:			✓ 🝰
iOS Device:			~ 😒
Start URL:	http://sahitest.com/demo/	training	~
		🕨 Run /	Record 💌
Verify Total	Total : 850	🕨 Run	/ Record
Amount		🕨 Run	
[End]		🎄 Debu	ug
User Logout			

In the Run/Record dialog select Run

We can notice that multiple times books are added according to the data we provided in the data file.

C A Not secure sa	hitest.com/demo/trair	iing/books.htm	1		\$
		All a	wailable bo	ooks	
	Title	In stock	Cost	Add quantity to cart	
	Core Java Ruby for Rails	5 12	Rs. 300 Rs. 200	-2 ab	
	Python Cookbook	7	Rs. 350	[-s]	
		Add	Clear Log	out	
			My Cart		
	Title	Quantity	Unit Co	st Total Cost	
		Grand Total	: 0		

Let's look at the logs:

Suites Suite Report Test Cases	Report Script Report	IS Code Coverage Report Su	ite Analysis						
Script Name: ShoppingCart.s.c	sv Auto Refresh 🗆								
Test Total Steps	Failures Errors Success I	Rate Time Taken	Node Load	Browser Inte	rmediate Statuses Fi	inal Status			
ShoppingCart.s.csv 48 Report Id: ShoppingCart_chrome_ Script Info Script Info	0 0 10 4248a3d900f350426c09	0 % 00:00:11 503 localhost e72061b17e9eccf1 Compare	:9999 -1 Chron 2 Logs	ne 112.0.0.0	NA	SUCCESS			
Total Test Cases Passed Failed 1 1 0	d Success Rate								
Test Case Id	Description Time	Taken Intermediate	Final Status						
TC01 Shopping Verify cart Cart	total after adding 00 books	:00:10 NA 062 NA	SUCCESS						
Starting script Expand All Collaps	🛯 🗠 All Show Failed 🗆 Sh	ow All Images 🗹 Convert To	English 🗆						
		loadSahi	Shoppin	gCartScript.sah					
TC01 Shopping Cart		[Documentation]	Verify ca adding b	irt total after oooks					
	[ShoppingCartData.csv: edentials]	Cr User Login	usernan	ie:test	Password:Mgkł	KEQBU			
			usernam	ie	Password]			
			test		MgkKEQBU				
	[ShoppingCartData.csv: oksQuantities]	Bo [Repeat]							
			Q		Q1		Q2	Total	
			1		1		1	850	
			10		5		3	5050	
			0		0		0	0	
			-2		ab		-5	0	
		[End]							
		User Logout							
Stopping script									

Data driven steps are shown with yellow heading table and data is listed in the table below.

Clicking on data reveals the functions called using the data.

oksQuantities]	[Repeat]				
		Q	Q1	Q2	Total
		1	1	1	850
[-] AddProgrammingBooks(1, 1, 1)					
_click(_row("Core Java5Rs. 300")) [890 ms] [07:57: _setValue(_textbox("q"), 1) [125 ms] [07:58:00.04: click(row("Ruby for Rails12Rs, 200")) [242 ms] [0	59.916 PM] L PM] 7:58:00.283 PM]				
_setValue(_textbox("q[1]"), 1) [143 ms] [07:58:00. _click(_row("Python Cookbook7Rs. 350")) [232 ms]	426 PM] [07:58:00.658 PM]				
_setValue(_textbox("q[2]"), 1) [142 ms] [07:58:00. _click(_div("All available books TitleIn stockCostAdd qu	800 PM] antity to cart Core Java5F	Rs. 300 Ruby for Rails12Rs. 2	00 Python Cookbook7Rs. 350	")) [241 ms] [07:58:01.041 PM]	
_click(_button("Add")) [253 ms] [07:58:01.294 PM]					
[+] VentyTotalAmount(850)					

Editing and Maintaining Test Scripts

When a change happens in business use case or business scenarios, tester can add new requirements to the existing test case without disrupting other steps. In our above test case, let's say we must clear cart before logging out. User can do this step easily by adding step to the file by right clicking and clicking Insert row above option.



Let's add function Clear to clear the cart. User can select run/record option to record the mentioned unimplemented test step.

		Amount	
		[End]	
		Clear 🔻	
		User Logout 🔻	
	TC01	lass dans also se	
	Shoppi	Insert row above	
		Insert row below	itials]
		Insert column left	
		Insert column right	Quantit
		Remove row	
		Remove column	
		Undo	
		Redo	
)		Run / Record	
1		Data Drive	
>			

Sahi runs the implemented functions. If it encounters any function that is not yet recorded, it'll open Controller to record the function.

ahitest.com/demo/trair	ning/books.htm				Z Sahi Controller - Google Chrome		
annesteenn, derno, trair	ing/books.ntm				sahi.example.com/_s_/spr/controller8.htm		
All available books Title In stock Cost Add quantity to cart Core Java 5 Rs. 300 2 Ruby for Rails 12 Rs. 200 ab Python 7 Rs. 350 -5 Cookbook -4dd Clear Logout				sahi.example.com says Recording Keyword: Clear Script: sahi.sample.run.sah To proceed, click OK. To modify, click Cancel and enter details as requir	ed. ОК - Choose	Cancel	
						D	
					Evaluate expression Recorded steps	Diagnose [Jocs Editor
	Γ	My Cart					•
Title	Quantity	Unit C	ost	Total Cost			
	Grand Total:	0			Sahi-Browser ▼ → Append Highlight Appended: _assertExists_(textbox("total")); _assertVisible_(textbox("total")); assertEqual("1400", etValue(textbox("total"))):	Choose 🗸	History
							li.

We click on clear button to add it to steps list, next we verify if the form cleared or not using control + hover on any of the quantity input fields. And then use assert button to add an assertion to the script.

Record Create Fn Playback Clipboard	Info	Sa
Function: /** * Clear		
*/ function Clear() { _click(_button("Clear")); _assertEqual("", _getValue(_textbox("q"))); }		
 Function Parameters (more) \$Q 		
Pass by object Remove Duplicates Save Re-record Abort		

Sahi goes through the clear step and will logout on cart application and stops.

In	sert testcase	Load Sahi Library Function Details	Import Insert	Tags Column Insert	Comment Col	umn				Sai	mple scenario	
	А	В	С	D	E	F	G	Н	I	J	К	
1	TestCase	Data	Key Word	Argument 1	Argument 2							
2			loadSahi 🔹	sahi_sample_run.sah								
3			loadSahi 🔹	ShoppingCartScript.sah								
4												
5	TC01 Shopping Cart		[Documentation]	Verify cart total after adding books								
5		[ShoppingCartData.csv:Credentials]	User Login 🔍	username : test	Password : MgkKEQBU							
7		[ShoppingCartData.csv:BooksQuantities]	[Repeat]									
8			Add Programming Books	Q:1	Q1 : 1	Q2 : 1						
9			Verify Total Amount	Total : 850								
0			[End]									
1			Clear									
2			User Logout									
3												
4												
5												
6									Diana da Charle			-
.7									Playback State	15		
8									Testcases: 0/0), Scripts: 0/1 BP	ROWSER: chro	me
9									ShoppingCart.	S.CSV		
)												
1									SUCCESS on	chrome		
2									ShoppingCart.	s.csv		

In logs:

Clear]
_click(_button("Clear")) [343 ms] [09:38:19.431 PM]) 554 DM]

Sahi Accessor Repository

To enhance script manageability and facilitate centralized editing, storage, and management of accessors, we aim to segregate accessors from script file code. Sahi's accessor repository functionality comes in handy here which allows us to create accessor repositories automatically either during the recording or thereafter using a single click.

Going to our functions library file, click on Create AR button in navigation bar.

Create S	Guite 🔻 📕 Create AR 🕌 Find Usage 🔹 Playback 📕 Stop All 🗙 Clea	ar Break Points More 💌	$\Leftrightarrow \Rightarrow$	
Shopp	ingCart.s.csv c × ShoppingCartScript.sah c ×			
Q 🎽	Auto Format $f_{ m I\!\!R}$ Create Function $f_{ m I\!\!Q}$ Show Context			
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	<pre>/** * UserLogin * @param \$username - * @param \$Password - */ function UserLogin(\$username, \$Password) { _setValue(_textbox("user"), \$username); _setPassword(_password("password"), \$Password); _click(_submit("Login")); } /** * AddProgrammingBooks * @param \$Q - * @param \$Q - * @param \$Q2 - */ function AddProgrammingBooks(\$Q, \$Q1, \$Q2) { </pre>	Create AR Script Directory: Script File: AR File: Proceed Ca	C:/SahiPro/userdata/scripts/ ShoppingCartScript.sah	× × AR.sah ×

Script Directory:	C:/SahiPro/userdata/scripts/		~
Script File:	ShoppingCartScript.sah		
AR File:	ShoppingCartAR	✓ .ar.csv	~



ShoppingCart.s.csv c × ShoppingCartScr			ipt.sah c×	Shoppin	gCartAR.ar.cs	sv c sahi_s
6						
	А		В		С	D
1	Key		Value			
2	<pre>\$_TEXTBOX_USER</pre>		_textbox("user	")		
3	\$_PASSWORD_PASSWORD)	_password("pa	assword")		
4	\$_SUBMIT_LOGIN		_submit("Logir	ו")		
5	\$_ROW_COREJAVA5RS300	_row("Core Ja 300")				
6	\$_TEXTBOX_Q		_textbox("q")			
7	\$_ROW_RUBYFORRAILS12RS200		_row("Ruby fo Rails12Rs. 20	r 0")		
8	<pre>\$_TEXTBOX_Q1</pre>		_textbox("q[1]	')		
9	\$_ROW_PYTHONCOOKBO	OK7RS350	_row("Python Cookbook7Rs	. 350")		
10	\$_TEXTBOX_Q2		_textbox("q[2]	')		
11	\$_BUTTON_ADD		_button("Add"))		
12	<pre>\$_TEXTBOX_TOTAL</pre>		_textbox("total")			
13	<pre>\$_BUTTON_CLEAR</pre>		_button("Clear	")		
14	<pre>\$_BUTTON_LOGOUT</pre>		_button("Logo	ut")		
	1		1			

Changes in Implementation of Sahi Script

Let's say user want to change the accessor element for a particular element. To update the accessor, first method is to open the ar file and change the accessor(only change if value is known), second method is to put a break point at the line of element in sahi script and run the script in debug mode. During the run, script execution is stopped when break point is encountered. User can then change the element's accessor and the use Save AR button to save the AR to file.

4	\$_SUBMIT_LOGIN	_submit("Login")	
5	<pre>\$_ROW_COREJAVA5RS300</pre>	_row("Core Java5Rs. 300")	
6	^{\$_TEXTBOX O}	_textbox("q")	
7	\$ ROW RUBYFORRAILS12RS200	row("Rubv for	

2nd method:

10	1
19	<pre>function AddProgrammingBooks(\$Q, \$Q1, \$Q2) {</pre>
20	_click(\$_ROW_COREJAVA5RS300);
21 •	_setValue(\$_TEXTBOX_Q, \$Q); \triangleleft breakpoint
22	_click(\$_ROW_RUBYFORRAILS12RS200);
23	_setValue(\$_TEXTBOX_Q1, \$Q1);
24	click(\$ ROW PYTHONCOOKBOOK7RS350):

		н
Script Directory:	C:/SahiPro/userdata/scripts/	
Script File:	ShoppingCart.s.csv	
Test Case 💙 :		~
Start Mode:	Browser	~
Browser:	chrome	· ·
Android Device:		∽ 🛸
iOS Device:		✓
Start URL:	http://sahitest.com/demo/training	~
 Run in parallel Run sequentially Run distributed (r 	Threads: 5	
 Run in parallel Run sequentially Run distributed (r 	Threads: 5	
 Run in parallel Run sequentially Run distributed (r Show advanced setting Reset 	Threads: 5 🛟 in single browser session multiple machines) gs Ant Target	Run -
 Run in parallel Run sequentially Run distributed (r Show advanced setting Reset Create 	Threads: 5 🔅 in single browser session multiple machines) gs Ant Target	Run -
 Run in parallel Run sequentially Run distributed (r Show advanced setting Reset * Create 	Threads: 5 \$	Run V Run Run Record
 Run in parallel Run sequentially Run distributed (r Show advanced setting Reset 	Threads: 5 \$	Run V Run V Run / Record

ahitest.com/demo/trai	ning/books.htm	า		Ĥ	sahi.e	xample.co	om/_s_/s	pr/contro	oller8.htm		
Title Core Java Ruby for Rails Python Cookbook	All a In stock 5 12 7 Add	Cost Rs. 300 Rs. 200 Rs. 350	Add quantity to cart 1 0 0 0 out	1 2 7 7 7 7 7 7	Record Set script Script Di Tile: Start Play tatemen setValue setPassw click(_sr click(_r aused at	Playbac t from file rectory: URL: Pause hts: ic_textbox(vord(_pass ubmit("Log w("Corp. 1: breakpoin" breakpoint	k Cliph :: C:/Sah Step : word("pa in")) wa5Rs 3 :: scripts	Stop "test") (Shopping) (Shopping)	Info data/scripts/ Diagnose , "****") gCartScript.si]]]	Sahi Pro 9.7
Title	Quantity Grand Total	My Cart Unit Co I: 0	ist Total Cost		setValue	(_textbox("q"), 1)	<u>/ snopping</u>	<u>gentscript.s</u>	aned1=21	1

Got to Record tab, and verify AR file path, enable AR repository checkmark, ctrol+hover on the required accessor. It's AR key and accessor will populate.

						Record P	layback Cl	ipboard Infe	0	Sa	hi Pro 9.7.0
Title	In stock	Cost	Add quantity to ca	urt		Click to s	tart recordin	g.			• •
Core Java 1	nover on Core Ja	va Rs. 300	1 3 hover on t	the input		Script Direc	ctory: C:/S	ahiPro/userdata	a/scripts/		~
Ruby for Rails	12	R s. 200	0			Script Nam	e: Shop	pingCartScript			
Cookbook	7	Rs. 350	0			AR File:	Shop	pingCartAR.ar.c	csv		🔹 🔽
	Add	Clear	out			Press CTRL	key and hove	er mouse over	r any eleme	ent.	
						AR Key:			C	lick on Anch	or Save AR
						Accessor:	cell("Cor	e Java")		2 ¥	Click
						Alternative	s: _cell("Co	re Java")		4 ~	Hilight
						Value:	Core Java		select a	ilternatives b	uttoget
		My Cart				Prefix:					Hover
		wiy Cart				Mode:	BROWSER	Sess	sion key:	sessionId	
Title	Quantity	Unit Co	t Total Cost			Assert L	ist Properties	🛃 Other	Actions:	Choose	~
inte	Quantity	Out Co.	i Iotai Cost								
						APTe List.					
					_						
\$_TEXTBO	DX_Q				S						
_textbox("q")			Ů							
textbox	("a")			- U							
_textbox	(4)			_Ľ.							
textbox	("q")			_							
_textbox	("q", _nea	r(_cell("Co	ore Java")))								
_textbox	(0, _near(_cell("Core	e Java")))								
PROMICER	,	Session k	ev. cossi	onId							

Select appropriate accessor and click on Save AR button.

	5100111, <u>_</u> 5_) 5p., 60								
sahi.example.com says									
Do you really	want to change t	he AR Entry?							
			ок	Cancel					
AK HIE.	Snoppingcan	LAK.dl.CSV		<i>©</i>					
Press CTRL key	y and hover mous	se over any eleme	ent.						
AR Key:	\$_TEXTBOX_Q			SaveAR					
Accessor: 🚓	_textbox("q", _ne	ar(_cell("Core Java	a"))) 🖞	Click					
Alternatives:	_textbox("q", _n	ear(_cell("Core Jav	/a"))) 🗸	 Hilight 					
Value:	1			Set					
Prefix:				Hover					
Mode:	BROWSER	Session key:	sessionId	t					

Click ok and go to Playback tab to stop the run.

Shopp	ingCartAR.ar.csv c×		
Q			
	А	В	С
1	Кеу	Value	
2	\$_TEXTBOX_USER	_textbox("user")	
3	\$_PASSWORD_PASSWORD	_password("password")	
4	\$_SUBMIT_LOGIN	_submit("Login")	
5	\$_ROW_COREJAVA5RS300	_row("Core Java5Rs. 300")	
6	\$_CELL_COREJAVA	_cell("Core Java")	_
7	\$_TEXTBOX_Q	_textbox("q",	
		_near(\$_CELL_COREJAVA))	
8	\$_ROW_RUBYFORRAILS12RS200	_row("Ruby for Rails12Rs.	

User can also change the key to match the new accessor of the element. Go to the script file and Ctrol+click on the key, it opens context menu.



User can see the name of the AR file, key and value. It shows option to edit the key.

Find what: <u>\$_TEXTBOX_Q</u> Find		
Replace with: Replace		
Exact Match: Function: Variable:	Change Japan	1 Tura sub Liu
File Pattern:		
(* = any string eg: *.sah or D:/test.sah,*test.s.csv)		
3 matches found in 2 files.	Accessor Repository (AR) Parame	eters
ShoppingCartAR.ar.csv	File Path:	
✓ 5: var \$ TEXTBOX 0 = textbox("a");	ShoppingCartAR.ar.csv 🗸	Open Inc
ShoppingCartScript.sah	Key:	
21: setValue(\$ TEXTBOX 0, \$0);	\$_TEXTBOX_Q	Edit AR Ke
✓ 42: assertEqual("" getValue(\$ TEXTBOX 0)):	Value:	
	_textbox("q")	
	Inline AR Key	

Find what: \$ TEXTBOX O	Find
	Penlace
Exact Match: Function: Variable:	Kepiace
File Pattern: (* = any string eg: *.sah or D:/test.sah,*test	t.s.csv)
3 matches found in 2 files.	
ShoppingCartAR.ar.csv	
✓ 5: var \$_TEXTBOX_Q = _textbox("q");	
ShoppingCartScript.sah	
21: _setValue(\$_TEXTBOX_Q, \$Q);	
42: _assertEqual("", _getValue(\$_TEXTBO	X_Q));
Updates	

Click on Replace button. And click ok.



Updated key

```
function AddProgrammingBooks($Q, $Q1, $Q2) {
    _click($_ROW_COREJAVA5RS300);
    _setValue($_TEXTBOX_JAVA,$Q);
    _click($_ROW_RUBYFORRAILS12RS200);
    _setValue($_TEXTBOX_Q1, $Q1);
    _click($_ROW_PYTHONCOOKBOOK7RS350);
    _setValue($_TEXTBOX_Q2, $Q2);
    _click($_BUTTON_ADD);
}
/**
```

Updated key in ar file.

				· ,	
1	<pre>\$_TEXTBOX_JAVA</pre>		_textbox("q",		
1			_near(\$_CELL	_COREJAVA))	
	A DOW DUDYEODDAY	4000000	(115) L (D 1 40D	

Sahi Testing Methods

Static Testing: Sahi is mainly used for automating applications. It also provides static testing to review source code of web applications.

Dynamic Testing: Dynamic testing is a technique that involves executing the software and observing its behaviour, often with the aim of identifying bugs or other issues.

Sahi Pro offers a wide range of features for dynamic testing of web applications. These include:

Recording and playback of test scripts: Sahi Pro allows users to record their interactions with a web application and play them back as test scripts, making it easy to automate repetitive testing tasks.

Cross-browser testing: Sahi Pro supports testing across a wide range of web browsers, allowing users to ensure that their applications function correctly across multiple platforms.

Data-driven testing: Sahi Pro supports data-driven testing, allowing users to test their application with different input data sets, to ensure that it functions correctly under different conditions.

Test reporting and analysis: Sahi Pro provides detailed reports and analysis of test results, allowing users to identify issues and track progress over time.

Black-box Testing: Black-box testing is a testing technique that focuses on the external behaviour of the software being tested, without looking at the internal workings of the code. In Sahi Pro, black-box testing can be performed by designing and executing test cases that focus on the functionality and user interface of the web application being tested.

Sahi Pro's record-and-playback feature, which allows users to record their interactions with the web application being tested and play them back as test scripts. This makes it easy to design and execute tests that focus on the user interface and functionality of the application.

Testing Levels

In Sahi Pro, unit tests can be written as functions that start with "test", such as "testUserNameMax50Characters" and "testPasswordAlphaNumeric".

For example, the "testPasswordAlphaNumeric" function could perform the following steps:

- 1. Generate a random password string of a length greater than 6.
- 2. Generate each character in the password string as a random alphanumeric character.
- 3. Verify that the operation of setting the password is allowed.

It is recommended to write comprehensive unit test cases and organize them in a single script for efficient testing.

To execute the tests, you can use the "_runUnitTests()" function. Running this function with no parameters will execute all tests. Alternatively, you can pass an array of specific test cases

to run only those tests. For instance, "_runUnitTests(["testUserNameMax50Characters", "testPasswordAlphaNumeric"])" will execute only the "testUserNameMax50Characters" and "testPasswordAlphaNumeric" test cases.

Operational Acceptance Testing can be done by running all the test cases in Sahi Pro using parallel test case execution. This testing verifies that a software application is ready to be deployed and used in a production environment.

Create test scripts: Create test scripts for each test case using Sahi Pro. These test scripts should include all the necessary actions and verifications required to validate the functionality of the application.

Execute the test scripts: Run the test scripts on the production-like environment. Ensure that the test environment is isolated from the actual production environment to avoid any unintended consequences.

Analyse the results: Analyse the test results to identify any defects or issues that need to be addressed before the application is deployed in the production environment.

Iterate and refine: If any defects or issues are identified, iterate, and refine the test cases and scripts until all issues are resolved.

Automatic correction and fix

Autoheal is a feature in Sahi Pro that enables automatic healing and recovery of failing scripts in web automation. It uses data from previous runs to identify alternate solutions when an element is not found, reducing failures and the need for manual intervention. Autoheal is only available for web automation in Sahi Pro.

Testing Types

Installation testing involves verifying that the software application can be installed successfully on different environments and configurations. Here are the steps to perform installation testing using SahiPro.

Compatibility testing involves verifying that the software application works correctly on different hardware, software, and network configurations.

Regression testing is a type of testing that is performed to ensure that changes made to the software application do not negatively impact existing functionality. In Sahi Pro, regression testing can be performed by re-executing existing test cases after making changes to the application and verifying that all existing functionality is still working as intended.

Steps to perform regression testing using Sahi Pro:

Identify the test cases: Identify the test cases that need to be re-executed as part of regression testing. These test cases should cover all the critical functionality of the application.

Create test scripts: Create test scripts for each identified test case using Sahi Pro. These test scripts should include all the necessary actions and verifications required to validate the functionality and performance of the application.

Execute the test scripts: Run the test scripts on the updated application and verify that all existing functionality is still working as intended.

Analyse the results: Analyse the test results to identify any defects or issues that need to be addressed before the updated application is deployed in the production environment.

Iterate and refine: If any defects or issues are identified, iterate, and refine the test cases and scripts until all issues are resolved.

Continuous testing is a software testing approach that involves automated testing throughout the software development lifecycle, from code changes to production deployment. Continuous testing ensures that issues are caught early in the development cycle, reducing the cost and effort of fixing defects later. In Sahi Pro, continuous testing can be achieved using the following steps:

Test automation: Write and automate tests using Sahi Pro for each new feature or change made to the application.

Continuous integration: Integrate automated tests into a continuous integration (CI) pipeline, such as Jenkins, Bamboo, or Travis CI, so that tests are run automatically after each code change.

Test reporting and analysis: Use Sahi Pro's test reporting and analysis features to generate test reports and identify defects and performance issues.

Test environment management: Manage test environments using tools like Docker, which provides a consistent and isolated environment for each test run

Collaborative testing: Collaborate with developers, testers, and other stakeholders to continuously improve the testing process and ensure that tests are up-to-date and relevant.

Destructive testing is a software testing technique that involves intentionally causing the software to fail to validate its robustness and resilience. In Sahi Pro, destructive testing can be performed using a variety of techniques, including:

Fuzz testing: Fuzz testing involves sending random and invalid input to the software to see how it handles unexpected input. This can help to identify potential security vulnerabilities, memory leaks, and other issues that could cause the software to crash or behave unexpectedly.

Stress testing: Stress testing involves putting the software under heavy load to see how it performs under extreme conditions. This can help to identify performance bottlenecks, memory leaks, and other issues that could cause the software to fail or become unresponsive.

Error injection: Error injection involves intentionally introducing errors, faults, or failures into the software to see how it responds. This can help to identify potential issues with error handling, recovery, and fault tolerance.

Boundary testing: Boundary testing involves testing the software at the limits of its input range, such as maximum and minimum values. This can help to identify potential issues with input validation and boundary checking.

Software performance testing is the process of testing software applications to measure their performance in terms of response time, throughput, scalability, and resource utilization under various workloads.

Security testing is the process of identifying vulnerabilities and weaknesses in a web application's security and ensuring that it can withstand malicious attacks.

Here are some tests we can perform security testing in Sahi Pro:

- 1. Authentication and Authorization Testing: Test the web application's authentication and authorization mechanisms by attempting to login with incorrect credentials, attempting to access pages or resources that require authorization, etc.
- 2. Injection Testing: Test the web application for SQL injection, XSS injection, command injection, etc. by injecting malicious input and observing the application's response.
- 3. Session Management Testing: Test the web application's session management mechanisms by attempting to steal another user's session, forcing the application to create a new session, etc.
- 4. Cross-site Scripting (XSS) Testing: Test the web application for XSS vulnerabilities by injecting scripts that can be executed by the browser, such as JavaScript or HTML.
- 5. Cross-site Request Forgery (CSRF) Testing: Test the web application for CSRF vulnerabilities by attempting to make unauthorized requests on behalf of the user, such as changing the user's password or transferring money.
- 6. Security Configuration Testing: Test the web application's security configuration settings, such as encryption, password policies, etc.
- 7. Vulnerability Scanning: Use Sahi Pro's vulnerability scanning feature to scan the web application for common vulnerabilities and weaknesses.

Testing Plugins

Users can run Sahi scripts using Maven. User can create a Maven project in Eclipse and set JDK path for running. Editing pom.xml file to the below code will enable run.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <proupId>sahi-examples</proupId>
  <artifactId>sahi-integration-example</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>sahi-integration-example</name>
<build>
     <plugins>
       <plugin>
          <artifactId>maven-antrun-plugin</artifactId>
          <version>1.7</version>
          <executions>
             <execution>
             <phase>generate-sources</phase>
                <configuration>
```

Changing **SahiInstallationDir, target name** and **antfile** according to your Configuration will enable Maven to build and run test cases. User can use <u>https://resources.sahipro.com/docs/using-sahi/playback-via-ant.html</u> to create antfile required.

Sahi offers integration with **Jenkins Continuous Integration system**. Sahi suites are executed periodically and CI system like Jenkins can trigger the test suites execution when code changes occur and committed to the code base. To playback test scenarios, Sahi Pro requires real browsers. However, many Jenkins/build machines run in server mode without a GUI, which makes it difficult to use Sahi Pro. As a result, end-users must either use headless browsers or execute Sahi scripts remotely to work around this issue.



Sahi integrates with JetBrains - TeamCity CI system. Configuration needs to be set before CI system can trigger test suite execution.

Details on setup are found here: <u>https://resources.sahipro.com/docs/using-sahi/integrate-with-teamcity.html</u>.

Sahi Integration Support

Sahi can be integrated with external tools using predefined URL calls. This document provides a step-by-step guide to using these URL calls for integration with Sahi. The guide is

divided into sections that correspond to the steps required for executing a script or suite from an external tool. Users can modify parameters in the guide to generate a URL specific to their environment.

The Sahi server typically runs on localhost:9999 by default. If you're using a Sahi server from another machine, you'll need to modify the host and port accordingly and then click the "Update" button.

Get Script Directory:

To get the script path specified in the sahi.properties file, you can use the following Request URL. You will get the scripts path as response as shown below.



["C:/SahiPro/userdata/scripts/"]

Get All Scripts Path in the given Directory:

http://localhost:9999/_s_/dyn/SahiEndPoint_scriptsList?suites=true&dir=C%3A%2Fsahipro%2Fuserdata%2Fscripts%2F

 C O D localhost9999/s_/dyn/SahiEndPoint_scriptsList?suites=true&dir=C%3A%2Fsahipro%2Fusedata%2Fscript \$	
The life is the set of	ຽ ≡
[1] C. Sanipro Userdials Scripts Geno 243.an, "C. Sanipro Userdials Scripts Geno	ta o o cripts : seripts ahipro ""C: ipts sahi"C: ipts sahi"C: ipts lex erdata data ripts lex C: alarcripts S erework S sahi arco con sahi arco con sahi arco con sahi arco sahi

Get Available Modes:

Request URL: <u>http://localhost:9999/_s_/dyn/SahiEndPoint_getStartWithOptions</u>

Response:

["GENERIC", "BROWSER", "ANDROID_BROWSER", "IOS_BROWSER"]

Request URL:	Response:						
<pre>http://localhost:9999/_s_/dyn/SahiEndPoint_getStartW ithOptions</pre>	["GENERIC","BROWSER","ANDROID_BROWSER","IOS_BROWSER"]						

Get All Available Browsers:

Request URL:	Response:					
<pre>http://localhost:9999/_s_/dyn/SahiEndPoint_getBrowse rList</pre>	["firefox","ie","chrome","edgenew","brave"]					
Test						

Get All Available Android Devices/Emulators

Name	Description	Value	е
refresh	If set to true, devices list will be refreshed.	true	~

Generate URL

Request URL:	Response:					
http://localhost:9999/_s_/dyn/SahiEndPoint_getAndroi dDeviceList?refresh=true						

Test

Get All Available iOS Devices/Simulators:

Running the Script/Suite:

To execute a suite or script, you can use a URL like this:

```
http://localhost:9999/ s /dyn/SahiEndPoint run?a=a
        &suite=C%3A%2Fsahi_pro%2Fuserdata%2Fscripts%2Fdemo%2Fsmall.suite
        &startWith=BROWSER
        &browserType=chrome
        &baseURL=http%3A%2F%2Fsahitest.com%2Fdemo%2F
        &androidDevice=XXXX
        &iOSDevice=XXXX
        &threads=5
        &isSingleSessionS=false
        &jsCodeCoverage=false
        &abortedRetryCount=0
        &failureRetryCount=0
        &logsInfo=
        &tags=
        &userDefinedId=2June2015_12_1_36_338
        &suiteTimeOut=0.0
        &isAutoHealEnabled=false
```

Users can visit this link to learn more about Sahi Pro integration with other software using REST APIs: <u>https://resources.sahipro.com/docs/using-sahi/sahi-integration.html</u>

Testing Overall

Automation is the strong point in Sahi Pro. User can just record any scenario on application under test and use playback to test it with different data.

Weakness: Sahi does one thing i.e., Automation and it's good at doing it optimally without overhead to maintain test scripts. Users must just maintain data related to the test cases correctly. No naming conventions are proposed in the Sahi docs itself.

Reports

Reports are generated after each execution. User can reach logs site to verify reports generated.



	Average Time Taken (sec)	Average II	me Taken By Bro	wsers		IZ = pw		100% 91-99% 76-90% 51-75% 26-50% 0-25%	Success chi	Percentage By Different Browser ome, 29 edgenew,	2		
Sui	tes			wser Type								Script Wise Su	ummary
									Scripts			Machine	-
		» <u>demo.sah</u>	http://sahit	chrome	Apr 28,	, 2023 11:32:43 AM	Apr 28, 2023 11:32:54 AM	00:00:10 748		Aborted:1	USER_ABORTED		0
		» <u>demo.sah</u>		chrome	Apr 28	, 2023 11:32:32 AM	Apr 28, 2023 11:32:34 AM	00:00:02 650		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 10:23:36 PM	Apr 27, 2023 10:31:37 PM	00:08:00 254		Aborted:1	USER_ABORTED		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 10:15:25 PM	Apr 27, 2023 10:23:19 PM	00:07:53 836		Aborted:1	USER_ABORTED		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 10:13:36 PM	Apr 27, 2023 10:14:58 PM	00:01:22 706		Aborted:1	USER_ABORTED		0
	» \$	ShoppingCartScript	http://sahit	chrome	Apr 27	, 2023 10:13:17 PM	Apr 27, 2023 10:13:23 PM	00:00:06 160		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 09:38:07 PM	Apr 27, 2023 09:38:25 PM	00:00:18 065		Passed:1	SUCCESS		0
	» <u>\$</u>	ShoppingCartScript	http://sahit	chrome	Apr 27	, 2023 09:37:46 PM	Apr 27, 2023 09:37:52 PM	00:00:06 049		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 09:32:11 PM	Apr 27, 2023 09:34:02 PM	00:01:50 436		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 09:31:42 PM	Apr 27, 2023 09:32:00 PM	00:00:18 079		Error:1	FAILURE		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 09:30:20 PM	Apr 27, 2023 09:30:56 PM	00:00:36 178		Failed:1	FAILURE		<i>°</i>
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 09:25:03 PM	Apr 27, 2023 09:29:34 PM	00:04:31 137		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 07:57:55 PM	Apr 27, 2023 07:58:13 PM	00:00:18 090		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 07:56:47 PM	Apr 27, 2023 07:57:05 PM	00:00:18 124		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 06:43:47 PM	Apr 27, 2023 06:43:59 PM	00:00:12 055		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 06:43:29 PM	Apr 27, 2023 06:43:41 PM	00:00:12 075		Passed:1	SUCCESS		0
		ShoppingCart.s.csv	http://sahit	chrome	Apr 27	, 2023 06:24:34 PM	Apr 27, 2023 06:40:59 PM	00:16:24 287		Passed:1	SUCCESS		0
		cart_test_suite.dd.csv	http://sahit	edgenew	Apr 25	, 2023 05:34:34 PM	Apr 25, 2023 05:34:48 PM	00:00:14 050		Passed:2	SUCCESS		0
		cart_test_suite.dd.csv	http://sahit	chrome	Apr 25	, 2023 05:33:15 PM	Apr 25, 2023 05:33:25 PM	00:00:10 048		Passed:2	SUCCESS		0
		<u>cart.s.csv</u>	http://sahit	chrome	Apr 25	, 2023 05:14:29 PM	Apr 25, 2023 05:14:37 PM	00:00:08 053		Passed:1	SUCCESS		0
		<u>sahi_sample_run.sah</u>	http://sahit	chrome	Apr 25	, 2023 04:49:13 PM	Apr 25, 2023 04:49:37 PM	00:00:24 102	1	Failed:1	FAILURE		0

Contributions

Naga Vara Pradeep Yendluri – Sahi Record feature, Data Drive, Sahi script research and its usage in automating various testing methods, BDTA execution.

Varuna Sri Budidi - Playback feature, Data driven test case execution

Yogesh – Sahi Economics & usage of automation and identifying elements in web applications

Hari Prasad – Sahi History, BDTA in Sahi

Sreenivas – Sahi Editor, BDTA applying in Sahi Scripts using scenarios.

Sumanth - Automation using Sahi. Parallel execution in Sahi using Data Driven Suites

Vamshi – Logging in Sahi, creating library functions in Sahi.

Ishitha – Suites in Sahi, suites execution.