https://qaautomater.vercel.app/

QAautoMATER

User Manual

]

CONTENTS

INTRODUCTION

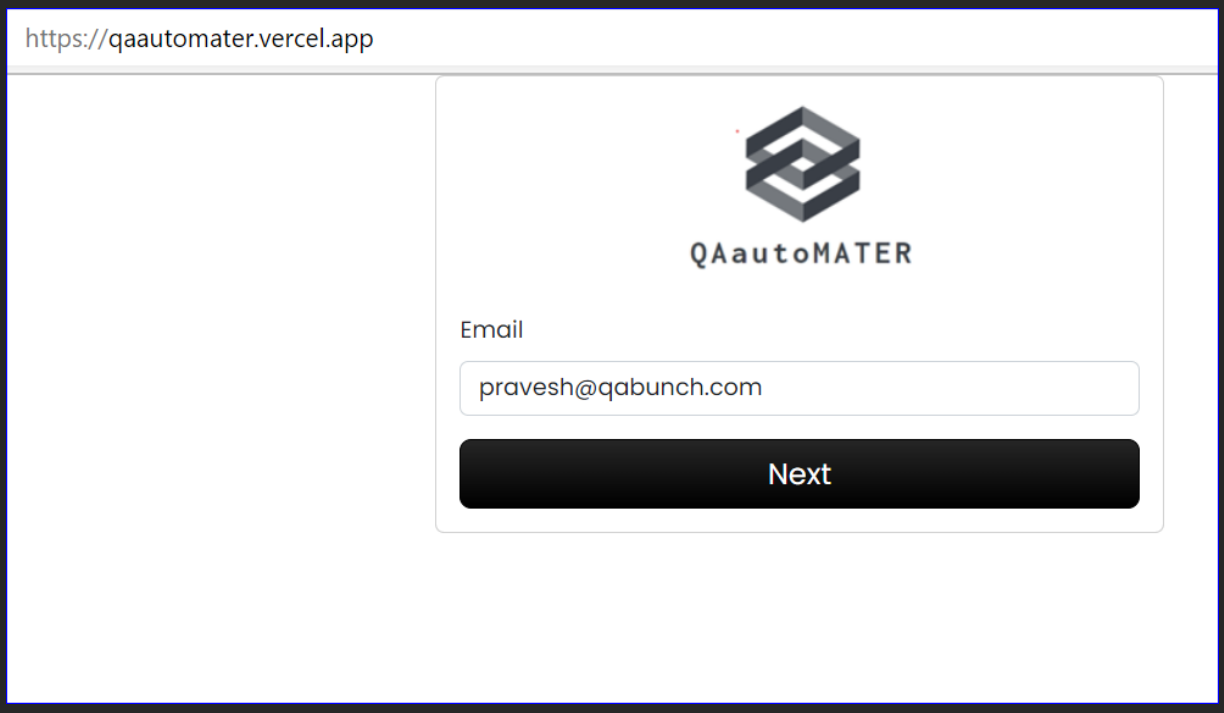
Welcome to the user manual for our automation testing software product QAautoMATER! Our software is designed to make your software testing process faster, more efficient, and more accurate. With our software, you can automate your tests using MI algorithm, reducing the time and resources required for manual testing, while ensuring that your software performs as expected. In this manual, we will provide you with a comprehensive guide to using our software, including installation instructions, basic and advanced usage, and troubleshooting tips. Whether you are a seasoned software developer or a newcomer to the field, this manual will help you to harness the power of our automation testing software and take your software testing process to the next level.

QAautoMATER is a complete codeless, ML Integrated, end to end QA Testing solution for web applications, Rest API and Hybrid mobile applications. This tool does not require the professionals using it to have any level of coding expertise as a pre-requisite. It is a UI driven tool, which can be used by any QA professional to start from writing manual test cases, then getting the automated test scripts/ test suite.

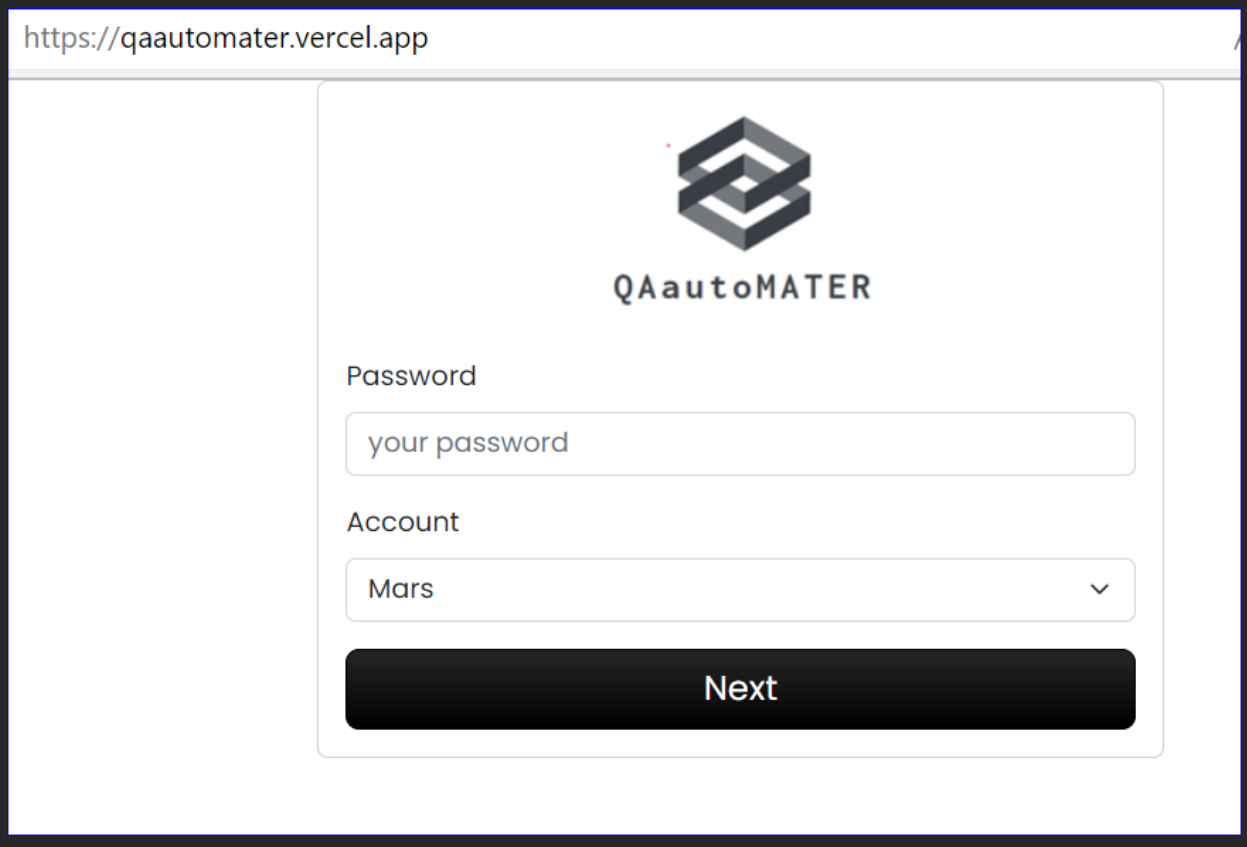
1. LOGIN

Login to the QAautoMATER on cloud Web Interface using QAautoMATER. (This tool can be access both on it is on cloud version as well the local version where it is installed in the client's environment. If you have the tool Installed in your environment, please use the respective URL to login.).

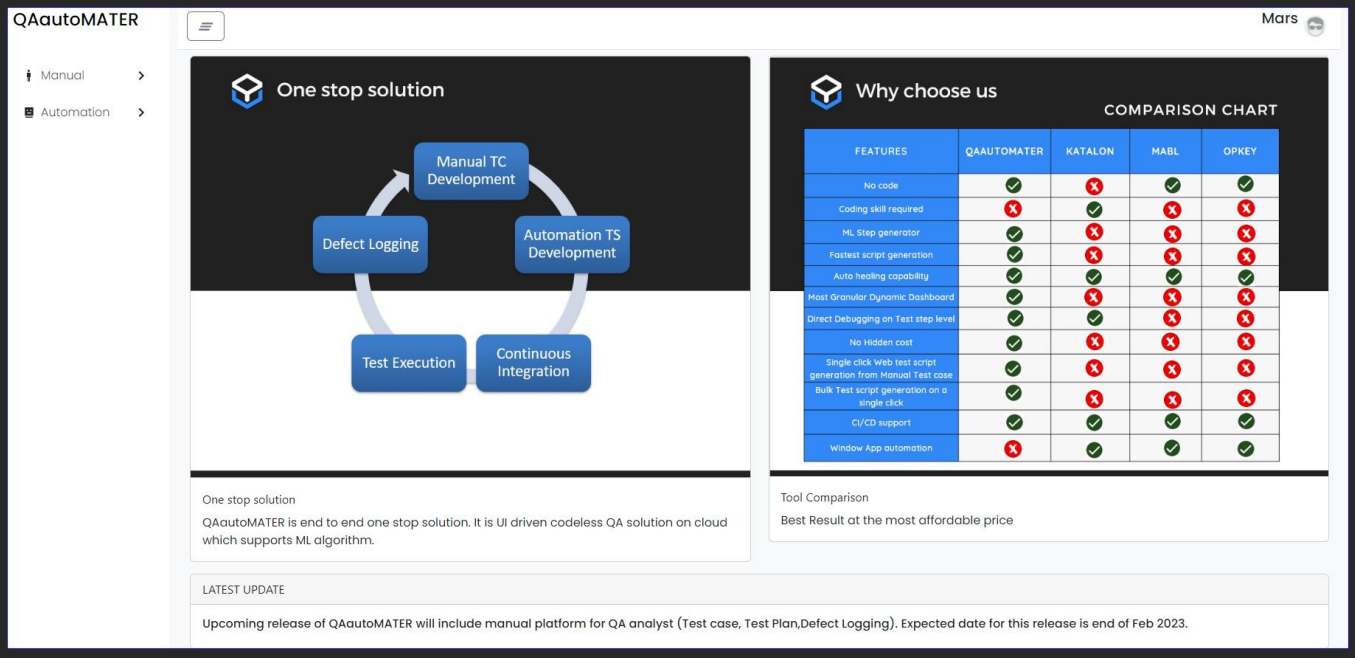
1.1. Provide your email Id registered with QAautoMATER. Click on Next.



1.2. Provide your password and choose the respective account from the drop-down list, which you are authorized to access.

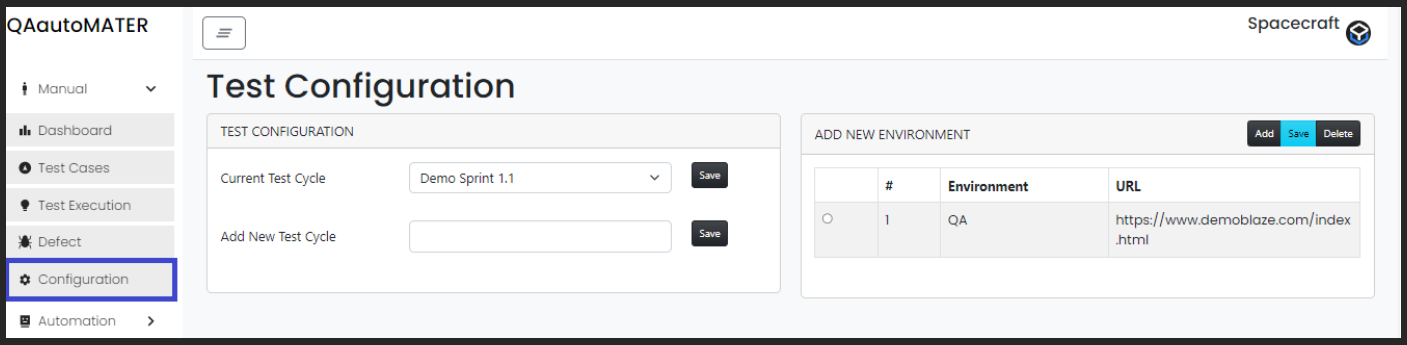


After successful Login, the user will be presented with a beautiful welcome page, which contains QAautoMATER lifecycle, an insightful comparison chart and the Latest updates regarding our tool. On the left hand side, QAautoMATER' s menu bar is present which contains paths for Manual and Automation. As an end-to-end QA solution provider we facilitate writing of manual test cases on this tool and then getting them automatically converted into automation test scripts. The steps for these will be discussed in the next few pages.



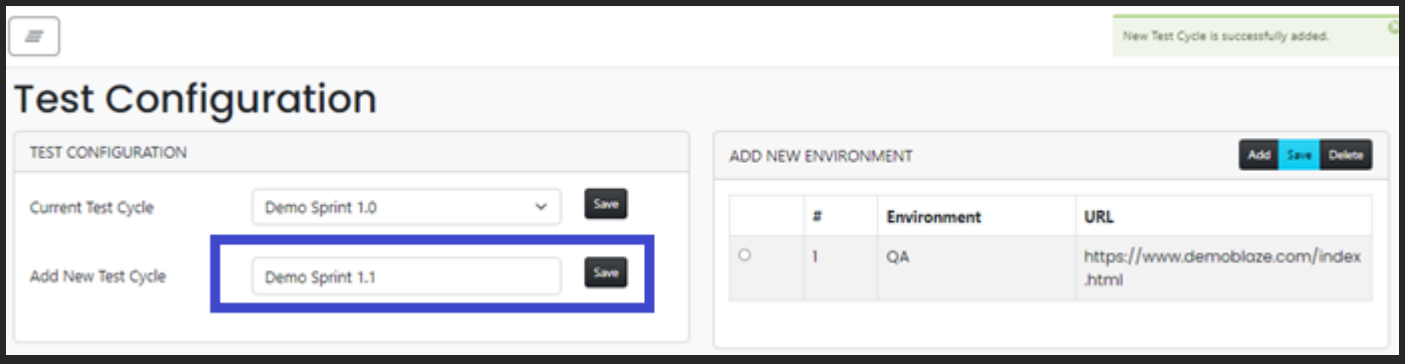
1.1 MANUAL - CONFIGURATION

From the left navigation panel, expand the "Manual" section and click the "Configuration" link.

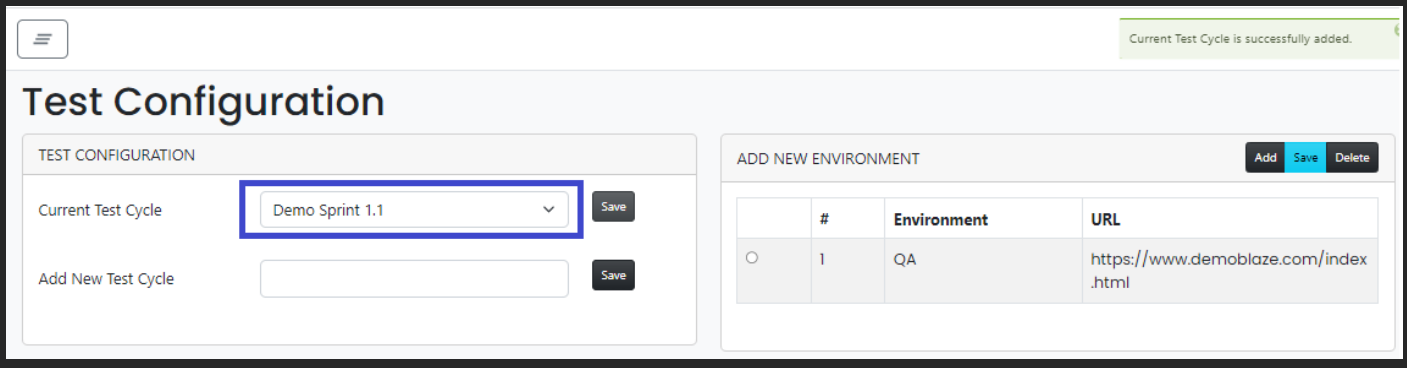


Add Test Cycle

On the configuration page, you can add a new sprint cycle by entering a sprint name in the 'Add New Test Cycle' field and clicking the 'Save' button

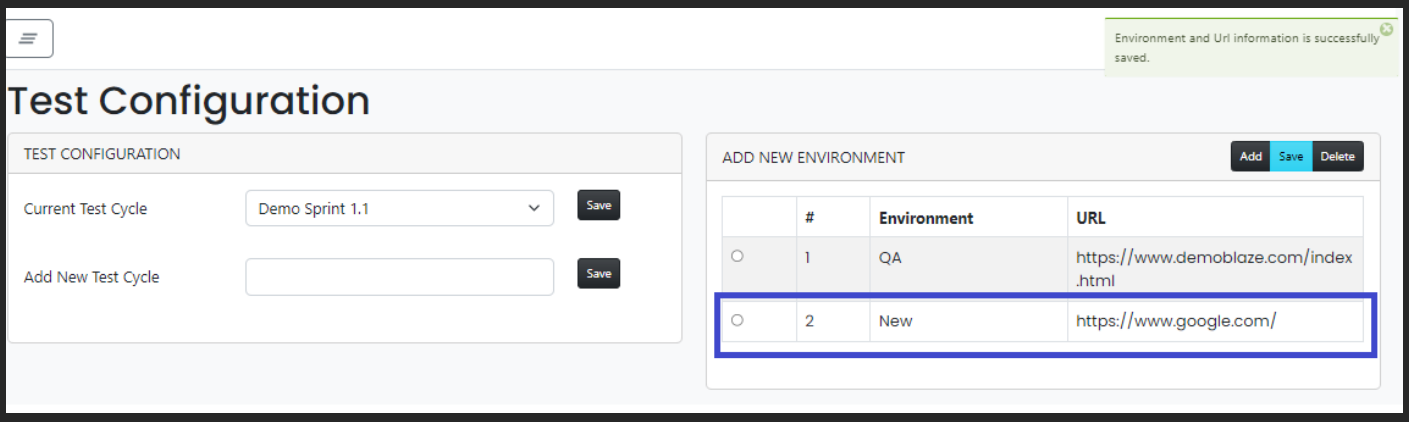


After successfully adding the test cycle, select your current cycle from the 'Current Test Cycle' dropdown and click 'Save'.

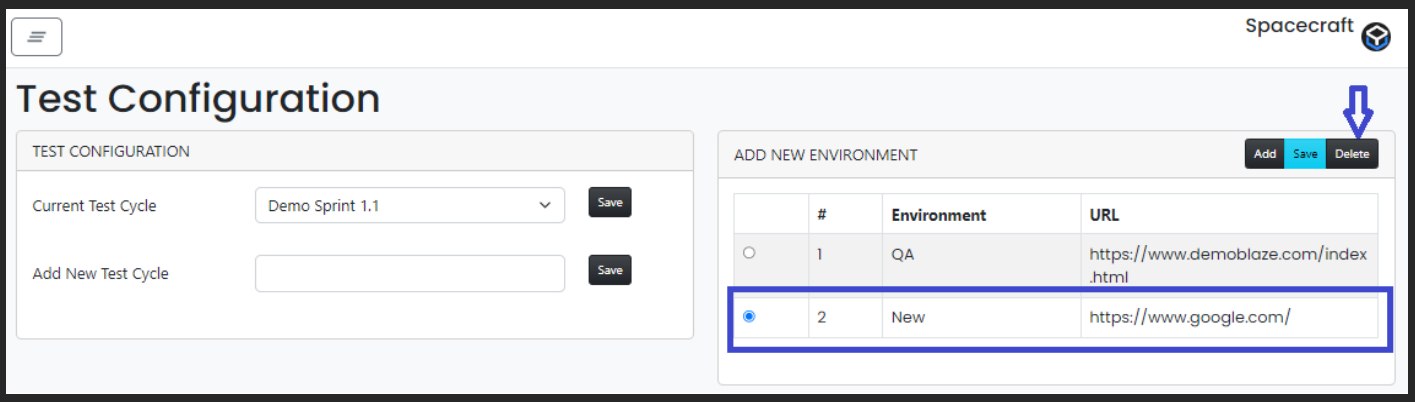


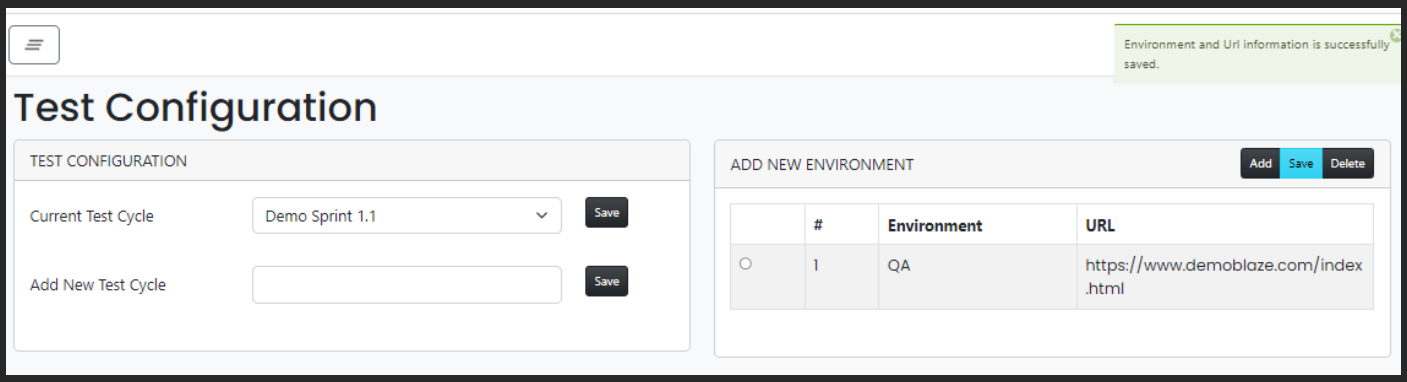
Add/Delete Environment

For adding a new Environment, click on the Add button, provide Environment name and URL details, and click the Save button.



To delete an environment, select the radio button in the environment column, click the delete button, and then click the Save button once the row has been deleted.



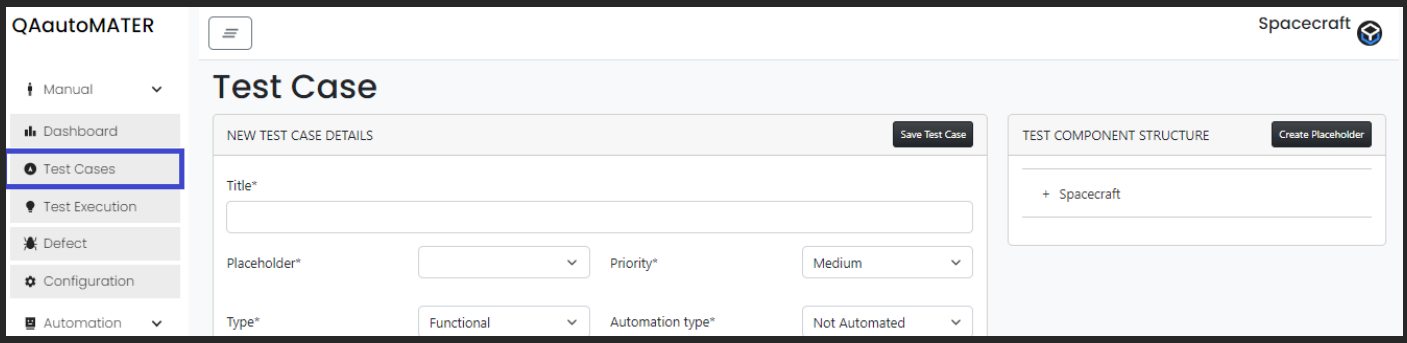


Recommendation:

It is recommended to update the current cycle at the end of each sprint. To do this, select the most recent sprint from the 'Current Test Cycle' dropdown and click 'Save'. This will ensure that the testing team is working on the most up-to-date sprint and that everyone is aligned on what tasks need to be completed. Updating the current cycle also helps to maintain a clear and organized testing process, which can ultimately lead to more efficient testing and higher-quality software.

1.2 MANUAL – TEST CASES

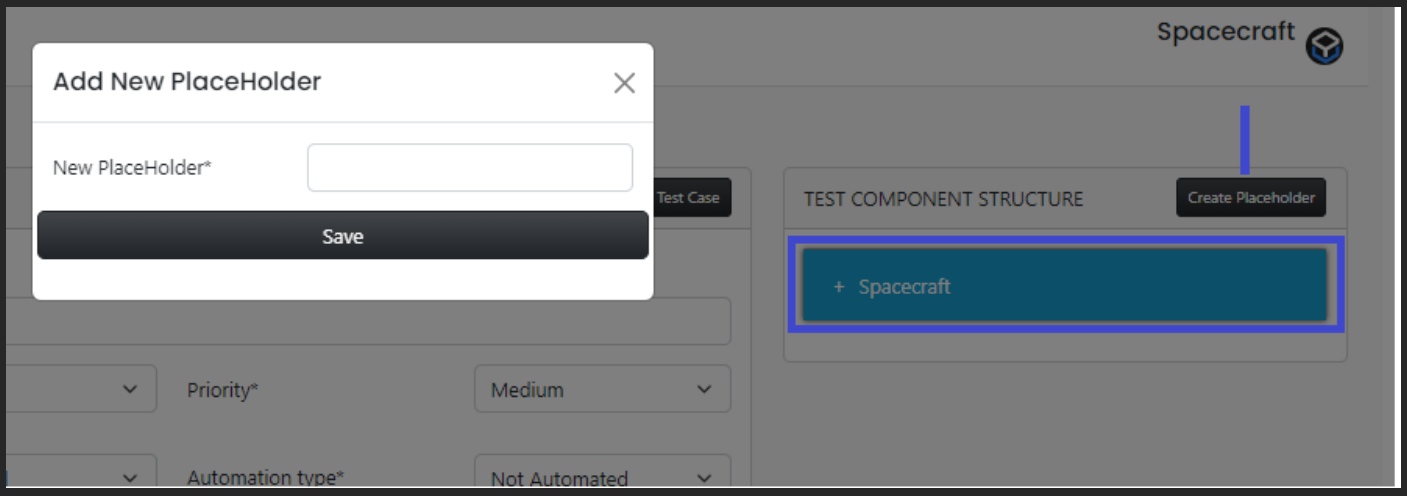
From the left navigation panel, expand the "Manual" section and click the "Test Cases" link.



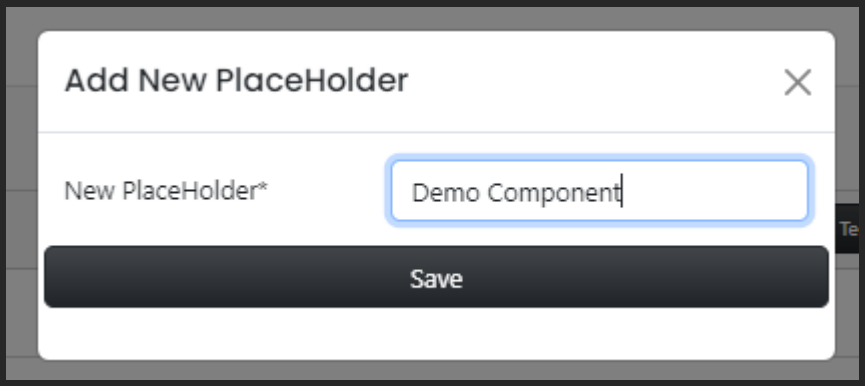
Create Placeholder or Component

To save your manual test cases, please create a folder named [PLACEHOLDER NAME] within this directory and save your test cases under that folder

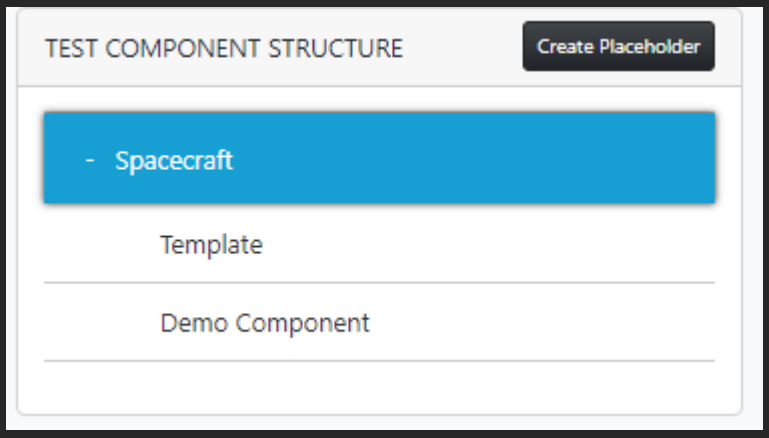
Click on Root folder (account) from the TEST COMPONENT STRUCTURE section and after that click Create Placeholder button



Provide New Placeholder name and click save button, after successful response newly created Placeholder can be found under root folder



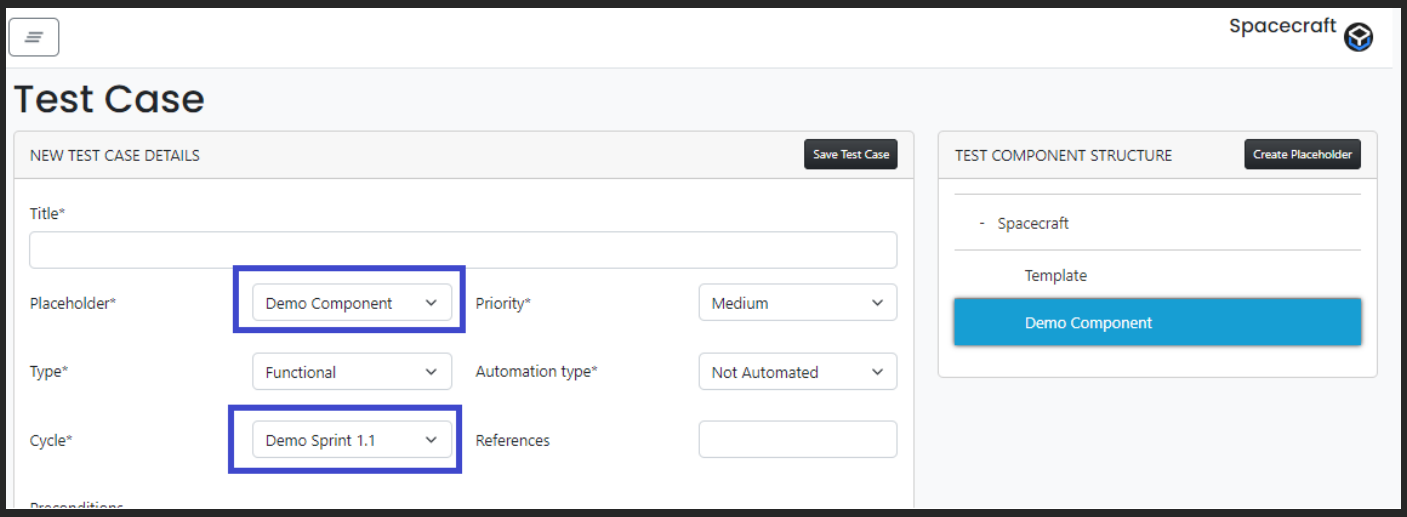
After expanding the root folder, new placeholder can be found.



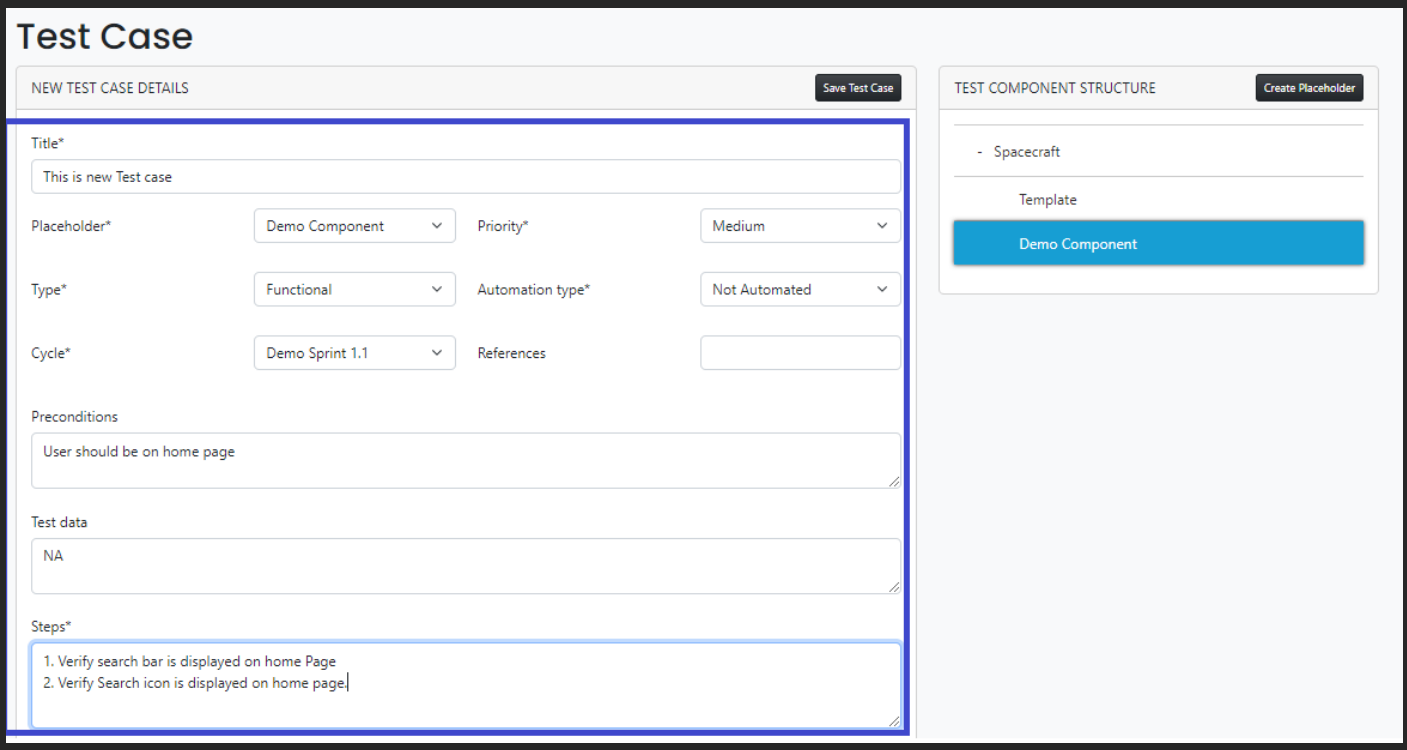
Please Note: We can also create folder inside the folder using same approach

Create New Manual Test Case

Select placeholder then you will get Manual test case from for development , Placeholder will automatically populated in manual test case form based on selection as well as Cycle will automatically populated based on selected current test cycle from the configuration page

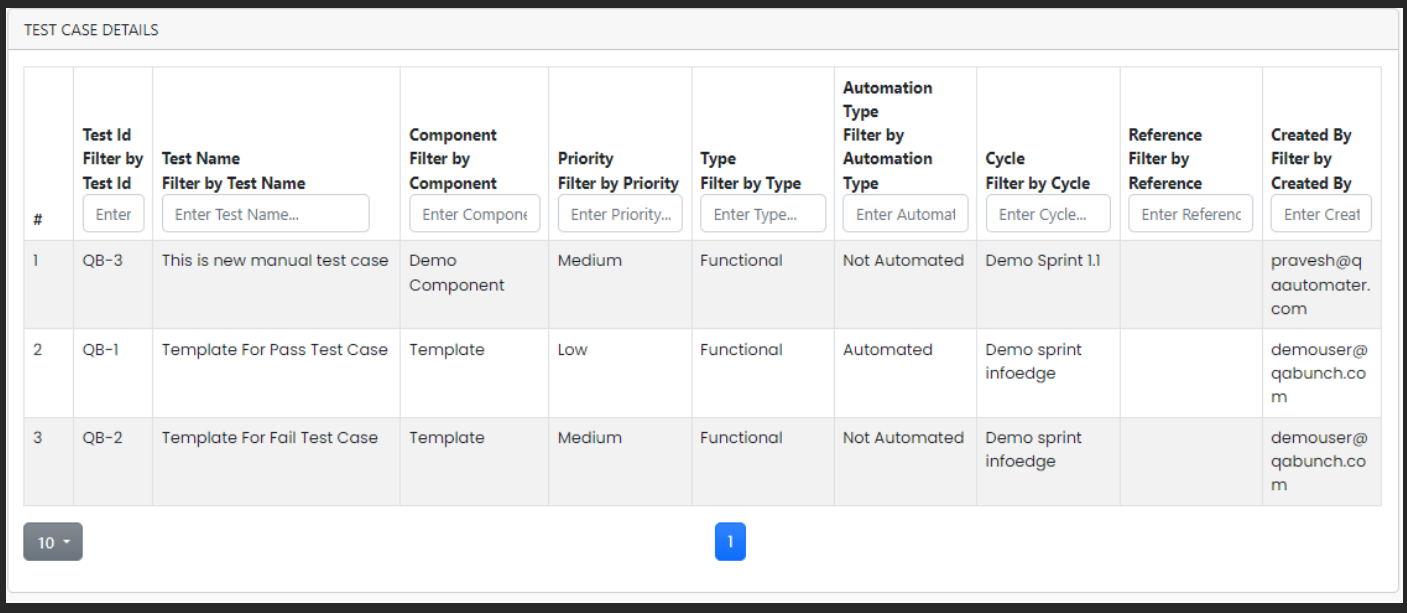


Add all manual test case attribute from the Manual Test case form and click Save Test Case button



Search Manual test case

Expand the Root folder and select placeholder, after loading the page, all manual test case from this placeholder will be displayed. For selecting root folder all manual test case will be displayed.

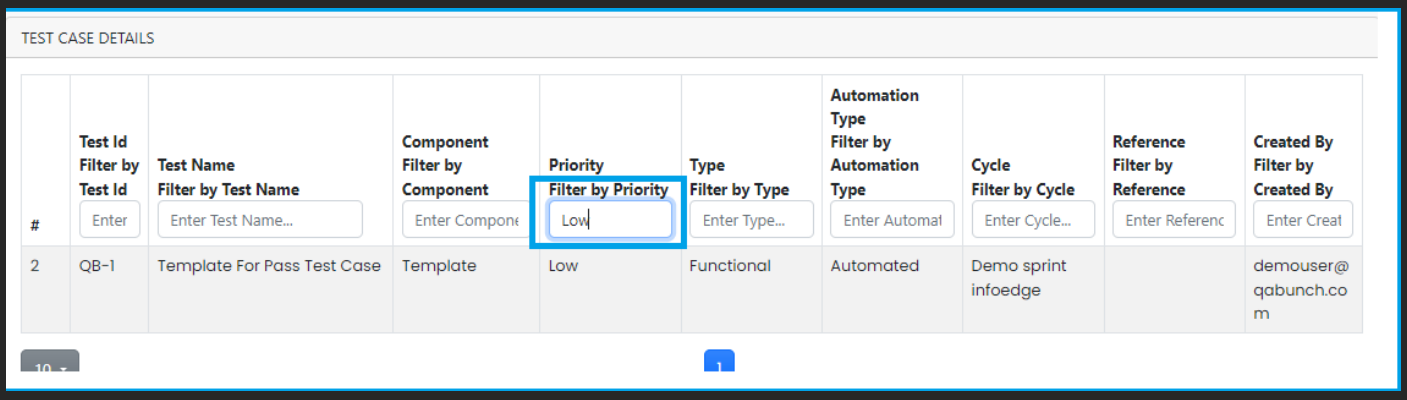


TEST CASE DETAILS

#	Test Id Filter by Test Id	Test Name Filter by Test Name	Component Filter by Component	Priority Filter by Priority	Type Filter by Type	Automation Type Filter by Automation Type	Cycle Filter by Cycle	Reference Filter by Reference	Created By Filter by Created By
	Enter	Enter Test Name...	Enter Compone	Enter Priority...	Enter Type...	Enter Automat	Enter Cycle...	Enter Referenc	Enter Creat
1	QB-3	This is new manual test case	Demo Component	Medium	Functional	Not Automated	Demo Sprint 1.1		pravesh@qaautomater.com
2	QB-1	Template For Pass Test Case	Template	Low	Functional	Automated	Demo sprint infoedge		demouser@qabunch.com
3	QB-2	Template For Fail Test Case	Template	Medium	Functional	Not Automated	Demo sprint infoedge		demouser@qabunch.com

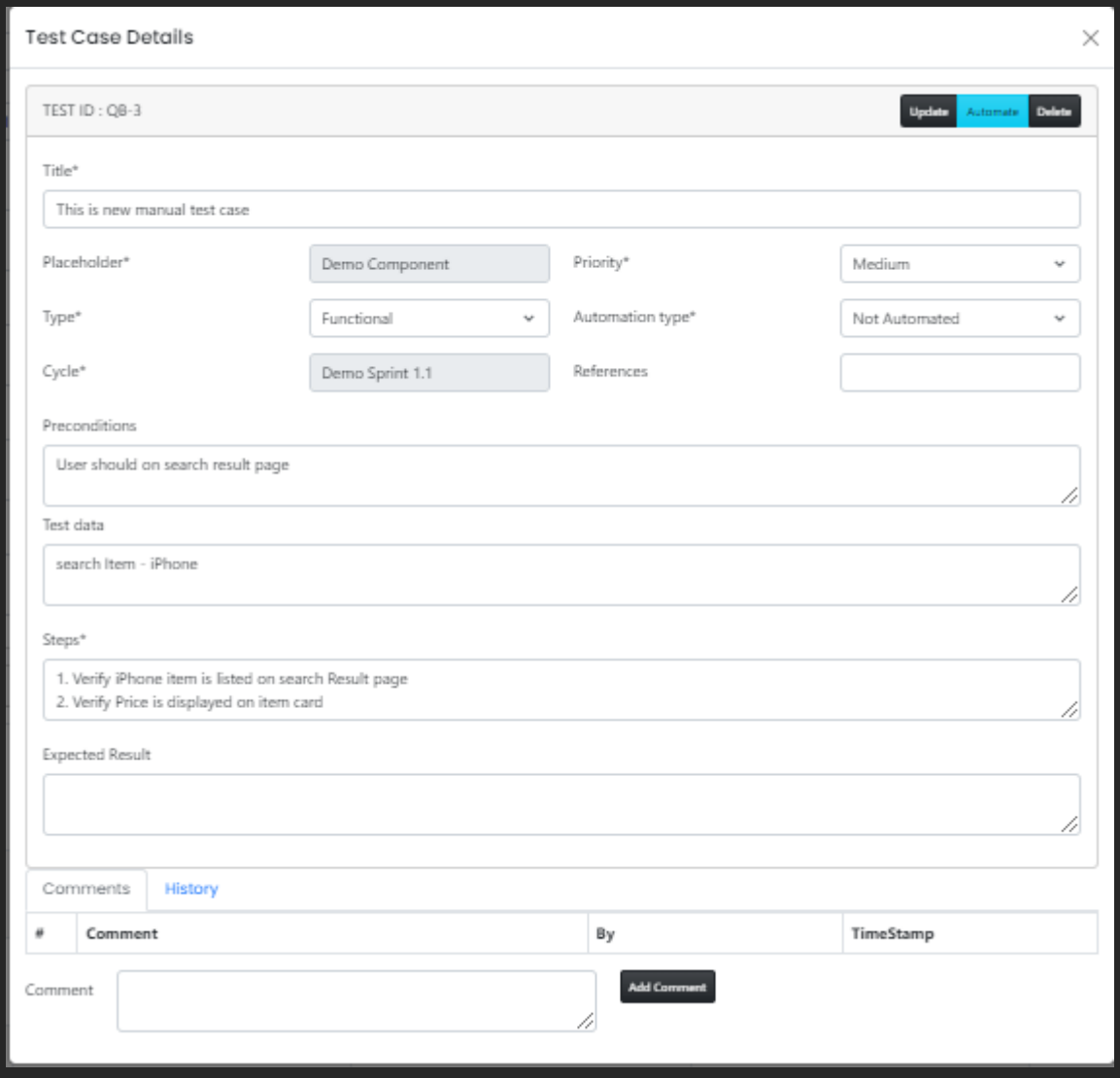
10 1

User can also search test case based on filter e.g. id, name, created By etc.



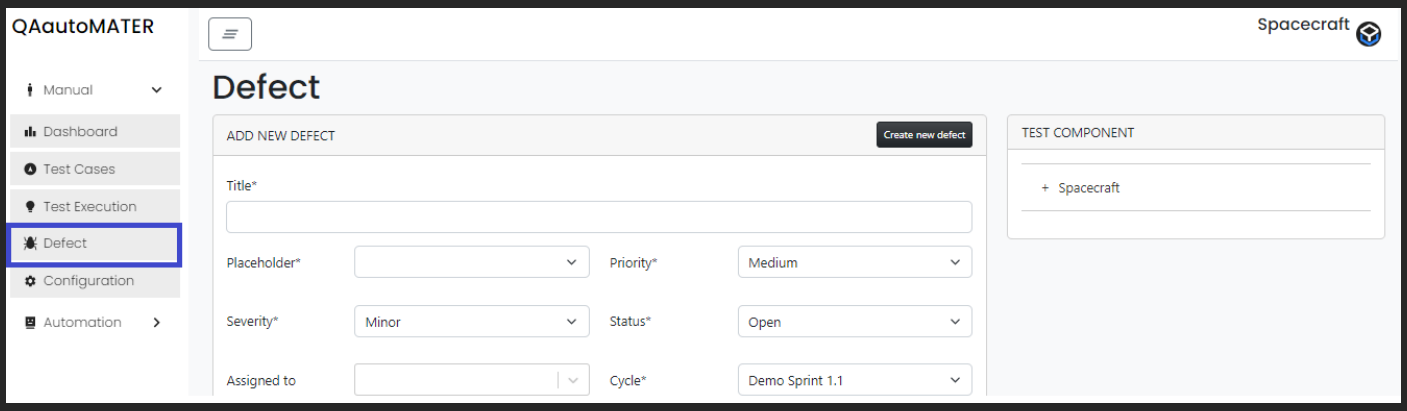
Update Manual test case

Find your manual test case from the test case details table and click the row then you will get the manual test case details dialog form, Update test attribute and click Update button.

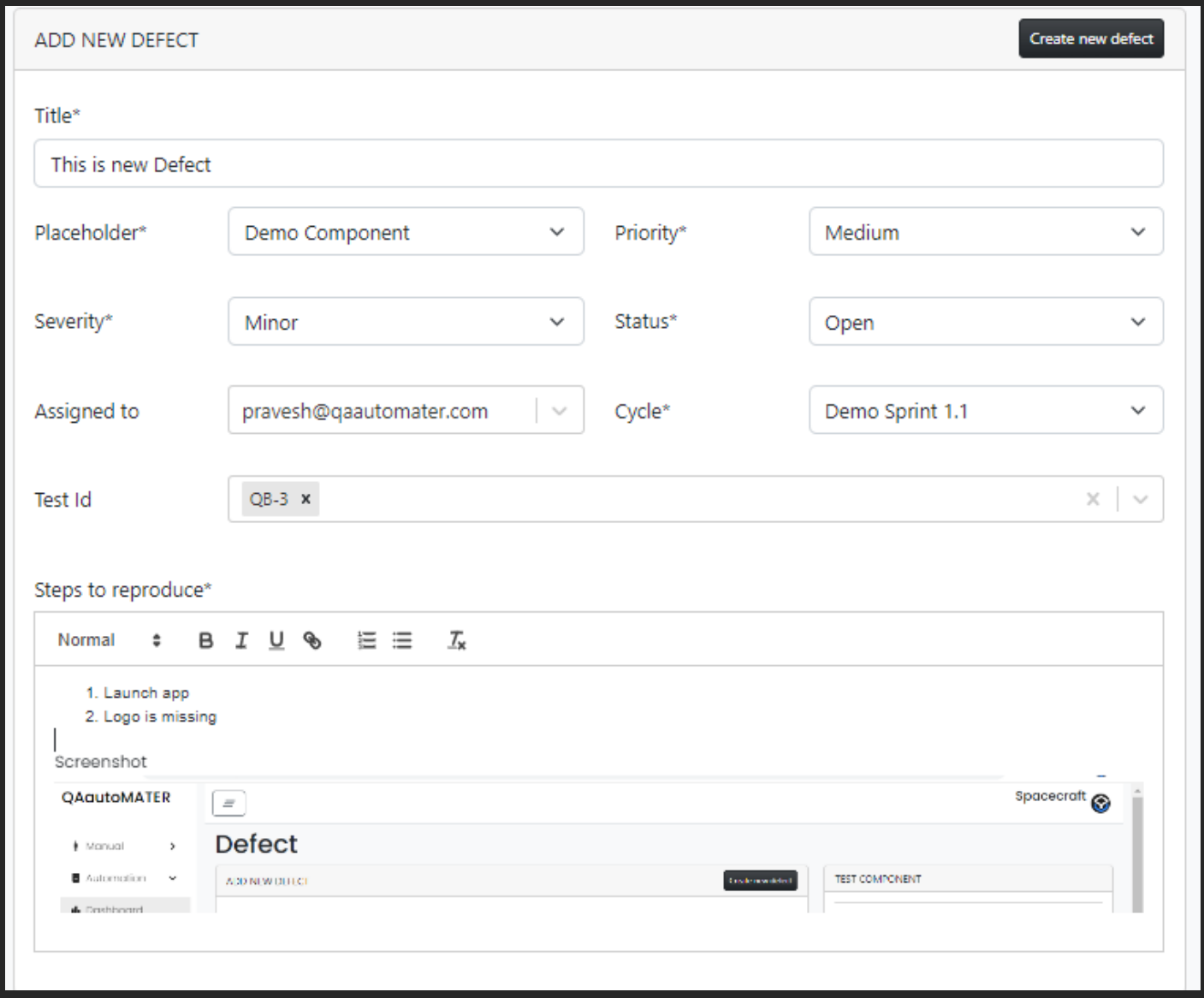


1.3 MANUAL – DEFECT PAGE

From the left navigation panel, expand the "Manual" section and click the "Defect" link.

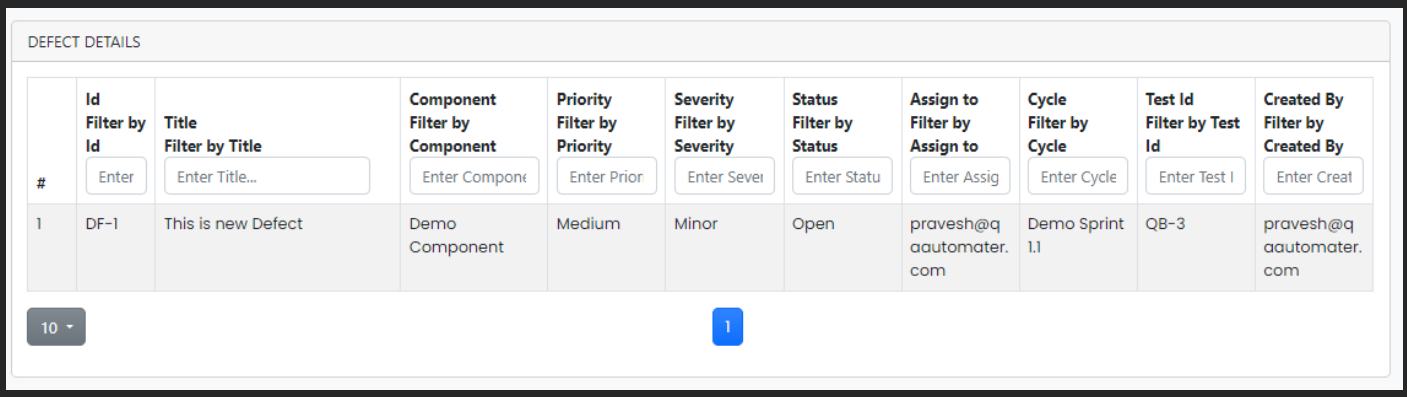


Add all information on defect form and click Create new defect button



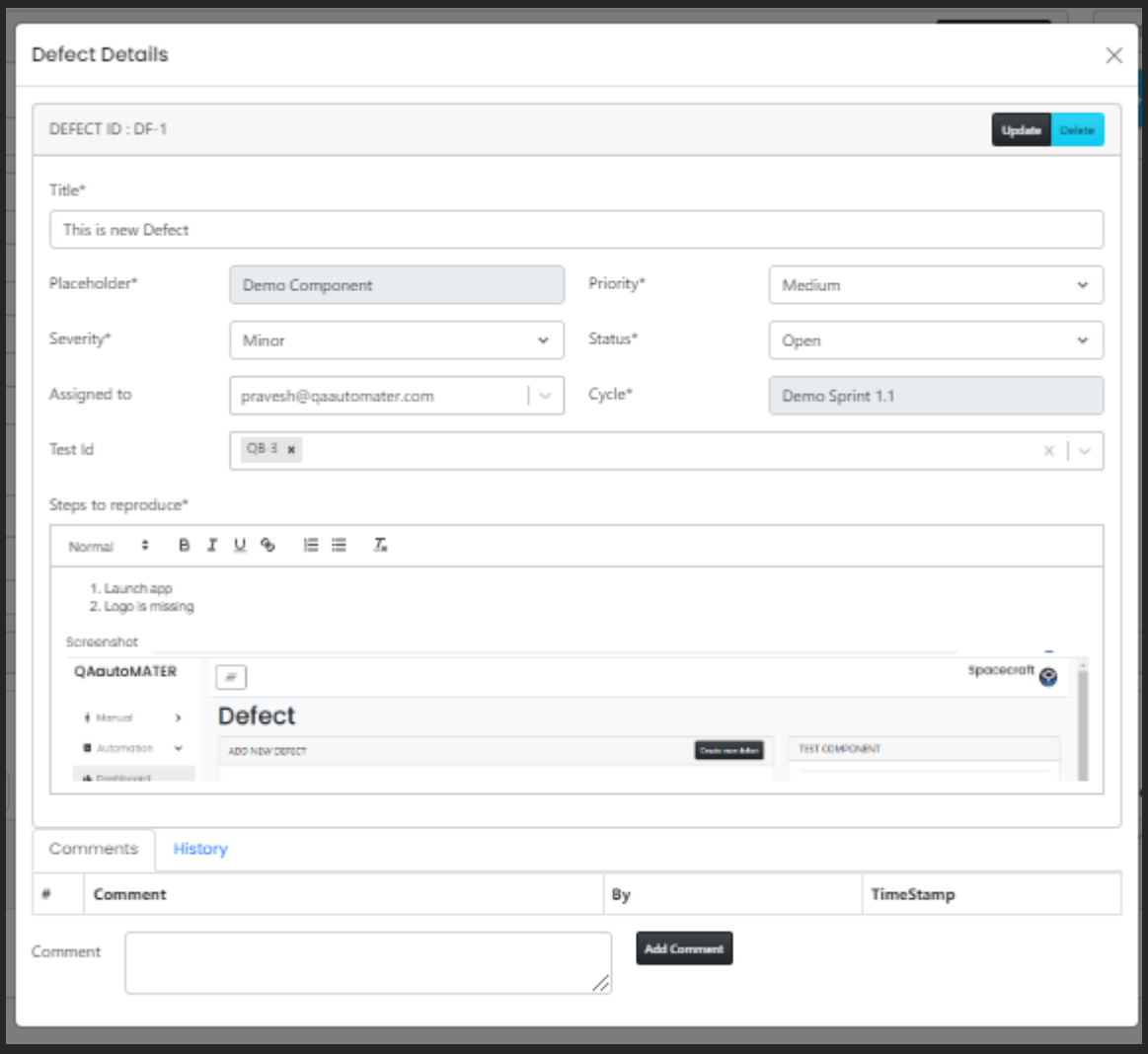
Search Defect

Expand the Root folder and select placeholder, after loading the page, all defect from this placeholder will be displayed. For selecting root folder all defect will be displayed.



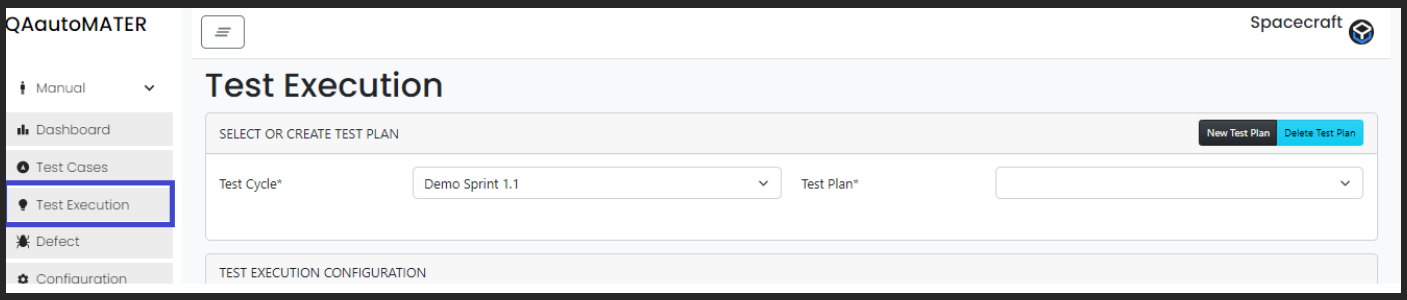
Update Defect

Find your Defect from the defect details table and click the row then you will get the defect details dialog form, Update defect attribute and click Update button.



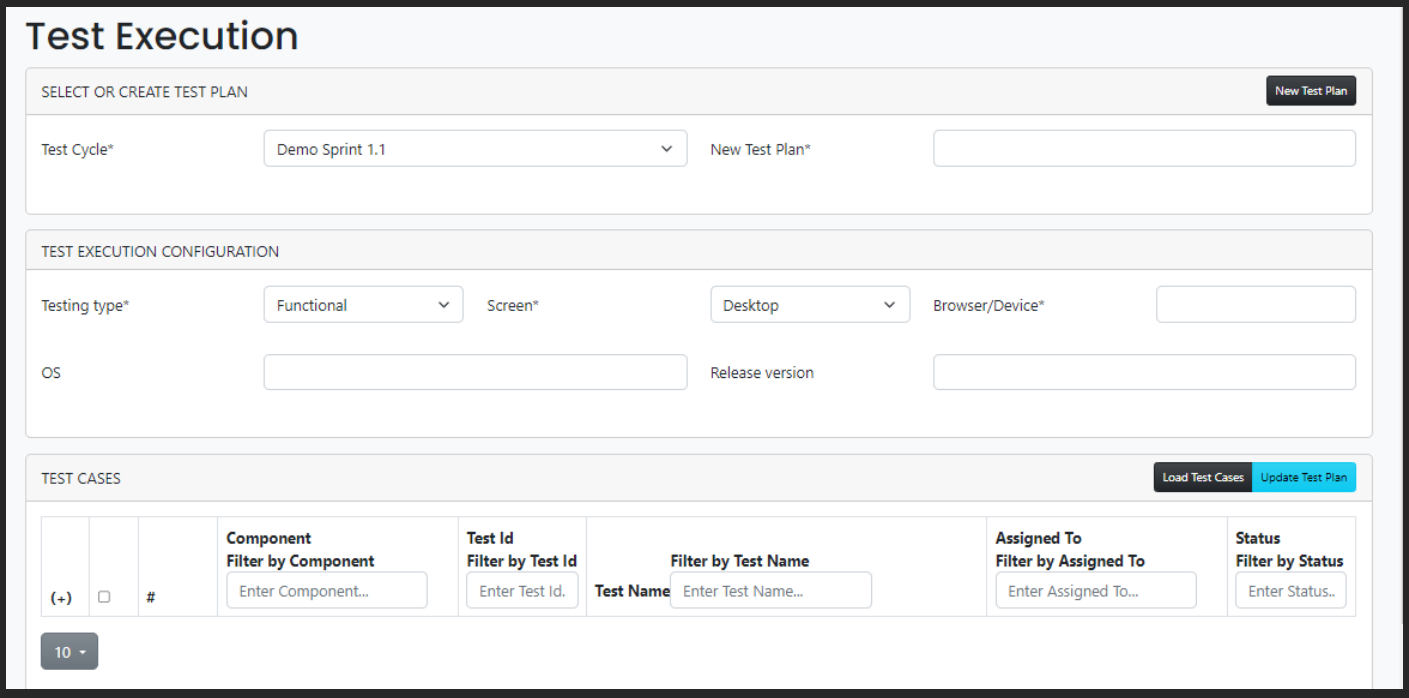
1.4 MANUAL – TEST EXECUTION

From the left navigation panel, expand the "Manual" section and click the "Test Execution" link.

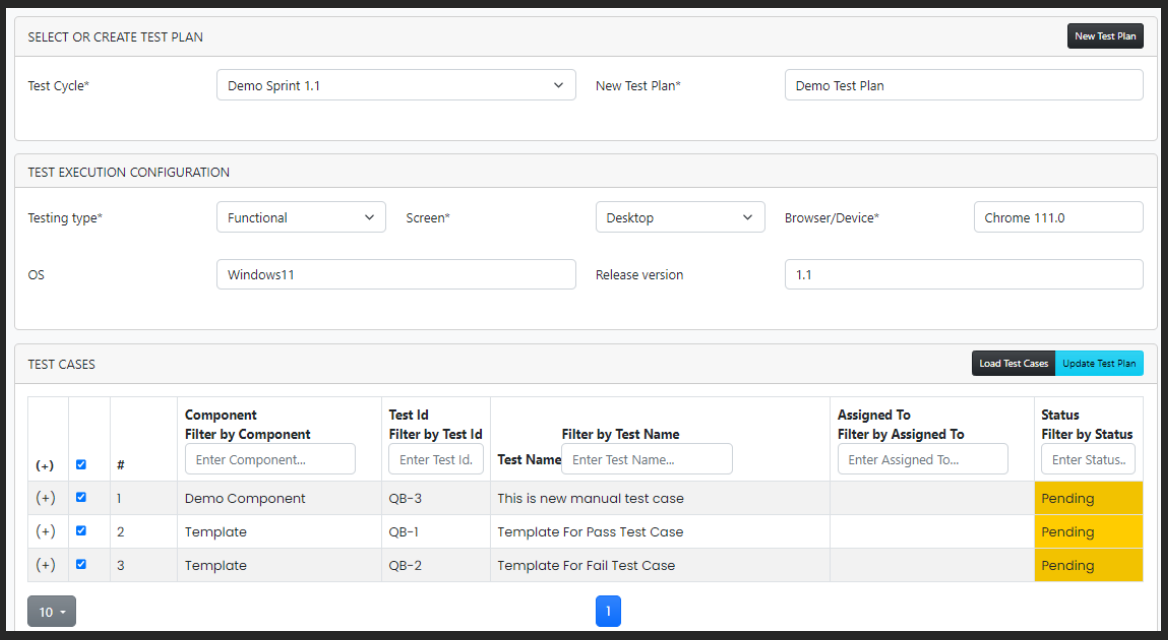


Create New Test Plan

From the test execution page, click on New Test plan button



Select test cycle, provide new Test plan name, Provide all test execution configuration after that click on Load Test case button Select the test case that you want to include in your test plan and click Update test plan button.

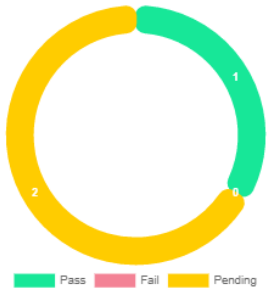


After creation of test plan, refresh the page and select test plan from the dropdown, all selected test case will be populated

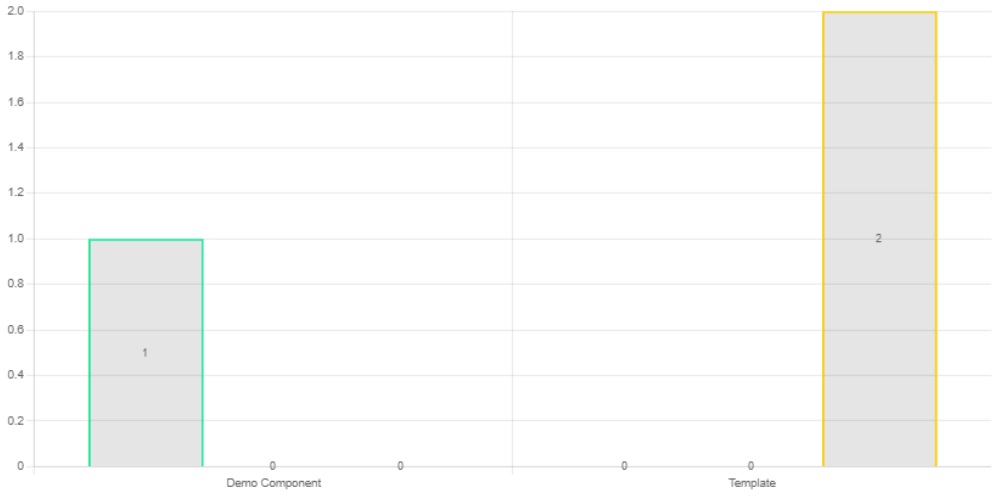
After that, you can update status for adding comment and document expand the test case, add comments and supporting document, and click Update Test Plan button

After updating test plan, graph will automatically updated.

EXECUTION STATUS



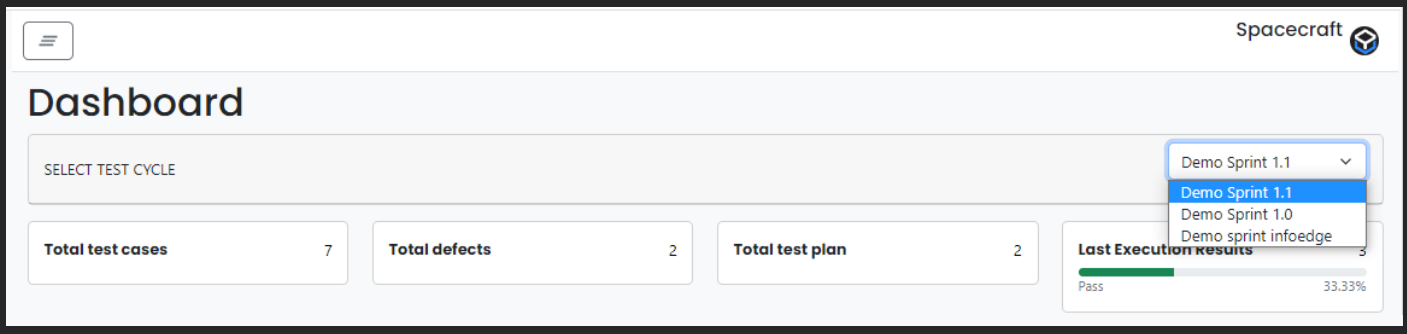
PASS FAIL COUNT MODULE WISE



1.5 MANUAL – DASHBOARD

By default dashboard gives the data for current test cycle, although user can view account and individual progress based on sprint cycle

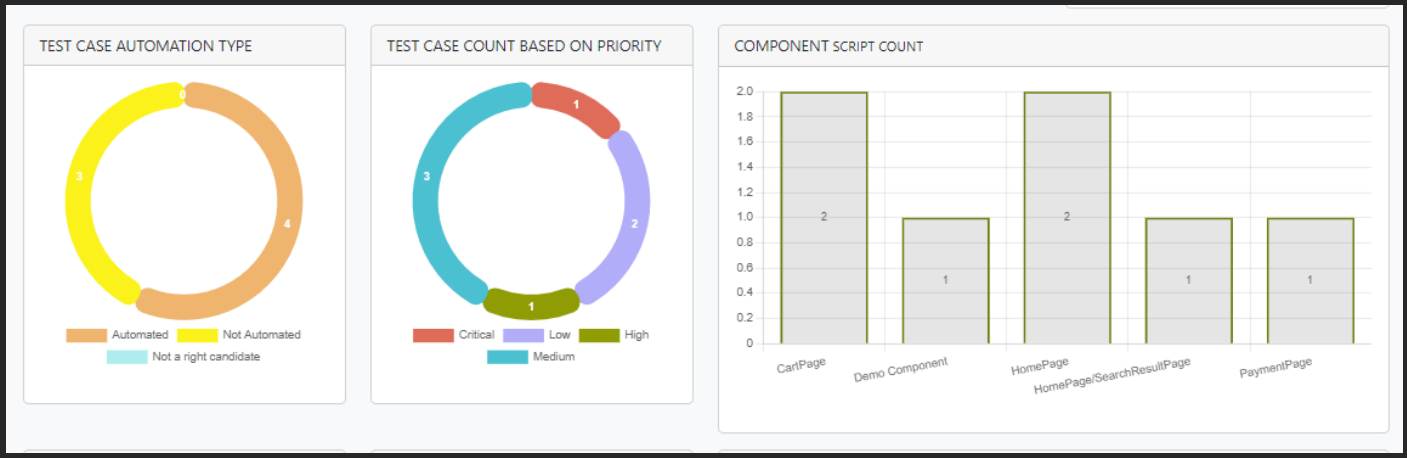
The widget displays data on the number of test cases, defects, and test plans created in the current test cycle, as well as the latest test plan results



The TEST CASE AUTOMATION TYPE section indicates the number of test cases that have been automated in the current test cycle.

The TEST CASE COUNT BASED ON PRIORITY section displays the number of test cases created in the current test cycle, categorized by their priority level.

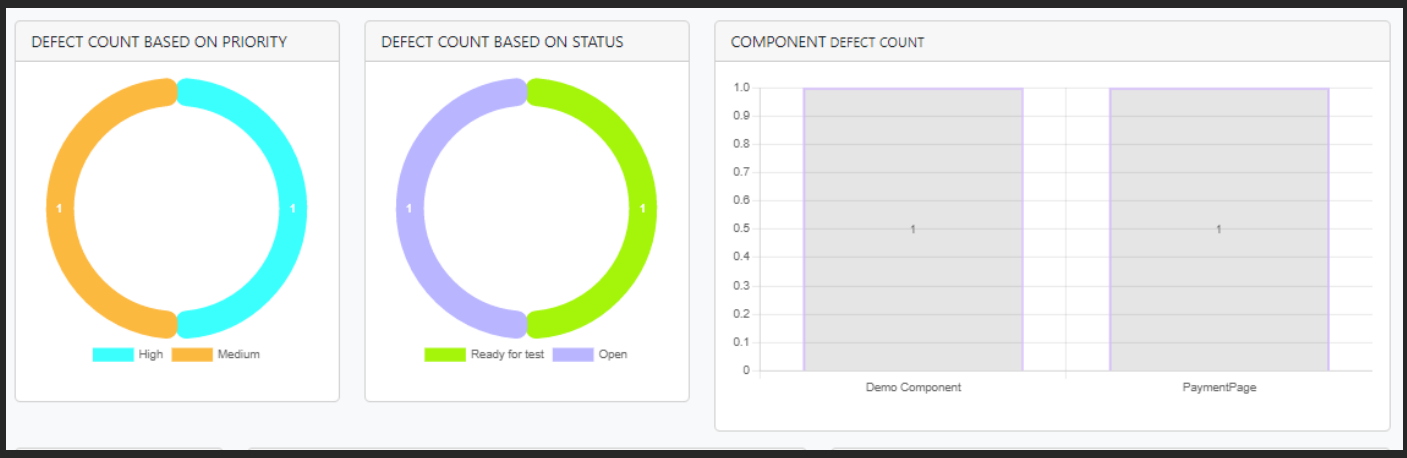
The COMPONENT SCRIPT COUNT section shows the number of test cases associated with each component.



The DEFECT COUNT BASED ON PRIORITY section shows the number of defects identified in the current test cycle, categorized by their priority level.

The DEFECT COUNT BASED ON STATUS section, which displays the status of defects in the current test cycle.

The COMPONENT Defect COUNT section shows the number of Defect associated with each component.



The TEST EXECUTION COUNT section displays the total count of test case executions for all test plans created in the current test cycle

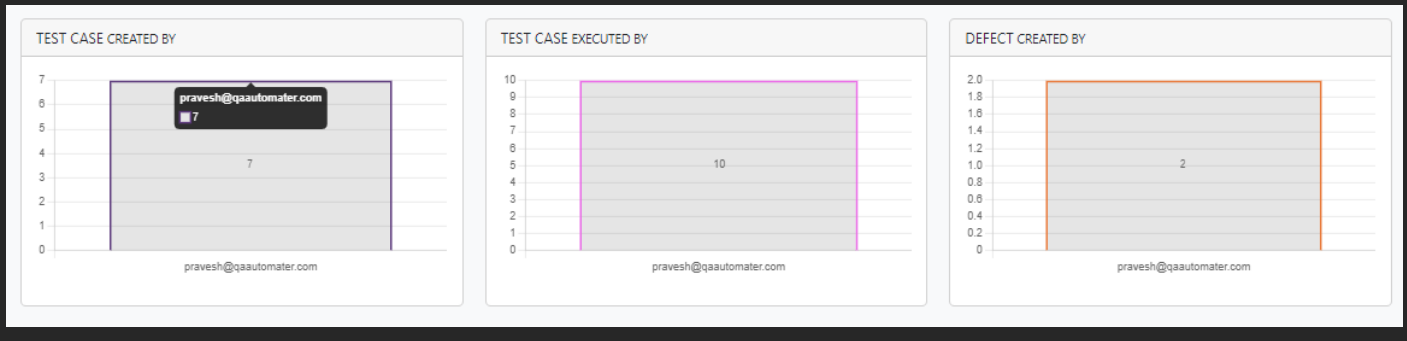
The PASS FAIL PENDING COMPONENT WISE section indicates the pass, fail, and pending counts for each component.

The PASS FAIL PENDING BY TEST PLAN section indicates the number of passed, failed, and pending test cases for each test plan.

The TEST CASES CREATED BY section indicates the number of test cases created by each individual in the current test cycle.

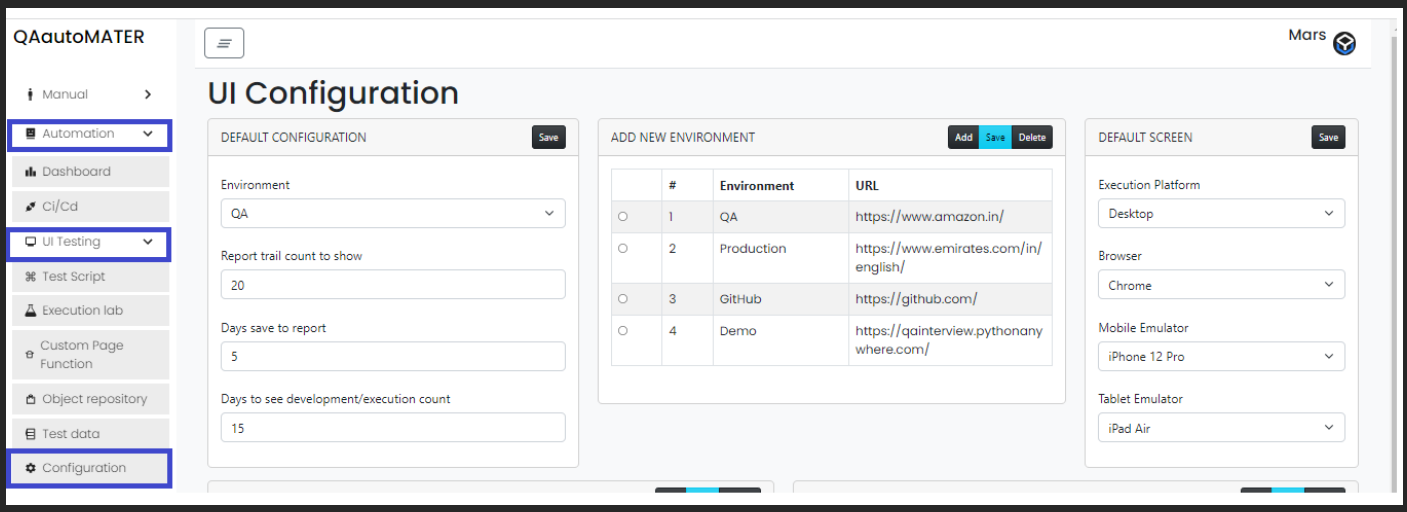
The TEST CASES EXECUTED BY section indicates the number of test cases executed by each individual in the current test cycle.

The DEFECT CREATED BY section indicates the number of defect created by each individual in the current test cycle.



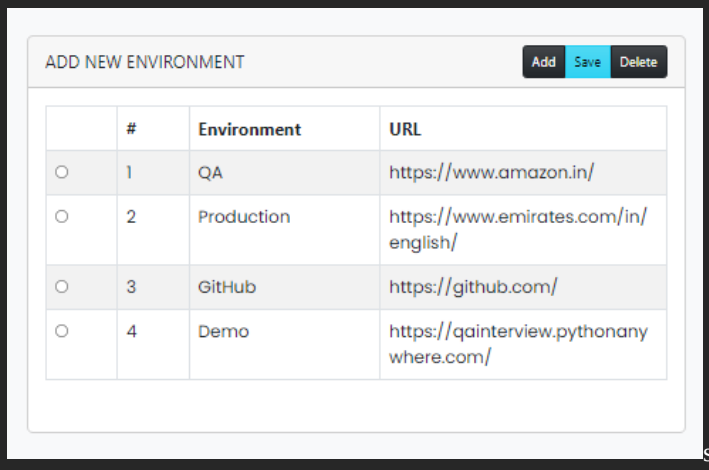
2.1 AUTOMATION – CONFIGURATION

From the left navigation panel, expand the "Automation" section and click the "UI Testing>Configuration" link.



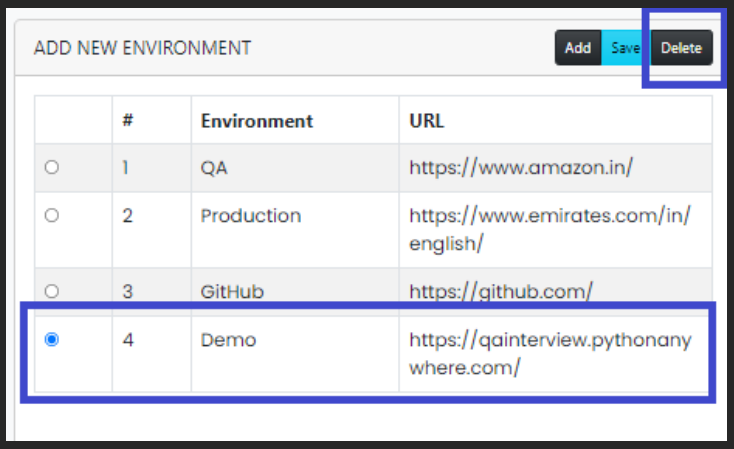
Add/Delete Environment

For adding a new Environment, click on the Add button from the 'ADD NEW ENVIRONMENT', provide Environment name and URL details, and click the Save button.

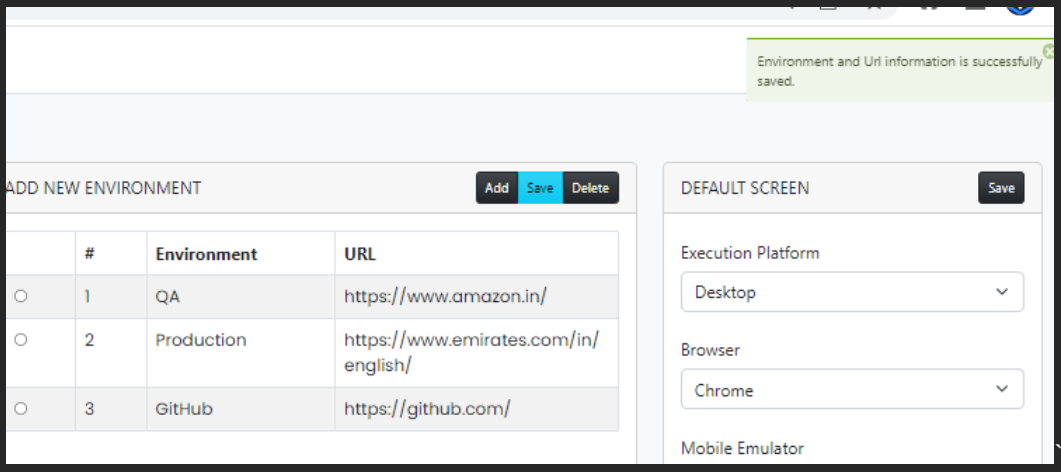


s

To delete an environment, select the radio button in the environment column, click the delete button, and then click the Save button once the row has been deleted.



After Clicking Save button you will get success notification and deleted environment will not displayed under ADD NEW ENVIRIONMNE table



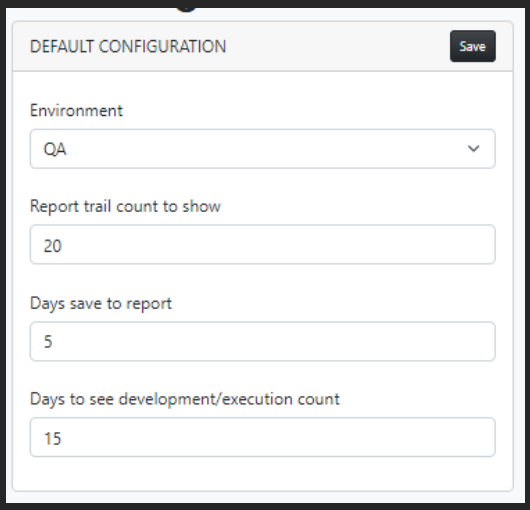
Default Configuration

Environment: You can select a default environment from the 'Default configuration' section, which will be displayed across all pages and will be pre-selected by default

Report Trail count to show: By setting the 'Report Trail' count to 10, for example, the dashboard page will display the execution history for the 10 most recent executions.

Days save to report: By setting, the 'Days to Save' report value to 10, for example, you will be able to see all the details of test executions that were performed within the last 10 days.

Days to see development/execution count: By updating the 'Days to See Development/Execution Count' value, we can view the total count of test script development and execution for a specified number of days on the dashboard page.



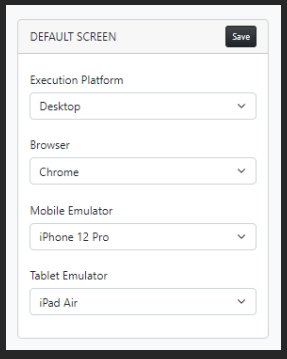
Default Screen:

Execution Platform: this tool allows for 3 execution platforms (desktop, mobile, and tablet), and the execution platform value is updated, then the selected mode for test script execution will be based on the updated platform value. For example, if the execution platform is updated to "mobile," then the test script will be executed in mobile mode. Similarly, if the execution platform is updated to "tablet," then the test script will be executed in tablet mode.

Browser: If a tool allows for selecting a browser for test execution and the browser value is updated, then the updated browser value will be preselected for the test case execution. For example, if the browser value is updated to "Chrome," then the test case will be executed in Chrome browser. Similarly, if the browser value is updated to "Firefox," then the test case will be executed in Firefox browser. The preselected browser value will be used for the test case execution until it is changed again

Mobile Emulator: when the test script is executed on a mobile platform, the mobile screen will be emulated according to the updated emulator value. For example, if the emulator value is updated to "iPhone X," then when the test script is executed on a mobile platform, the mobile screen will be emulated as an iPhone X

Tablet Emulator: when the test script is executed on a tablet platform, the mobile screen will be emulated according to the updated emulator value. For example, if the emulator value is updated to "iPhone X," then when the test script is executed on a mobile platform, the tablet screen will be emulated as an iPad air



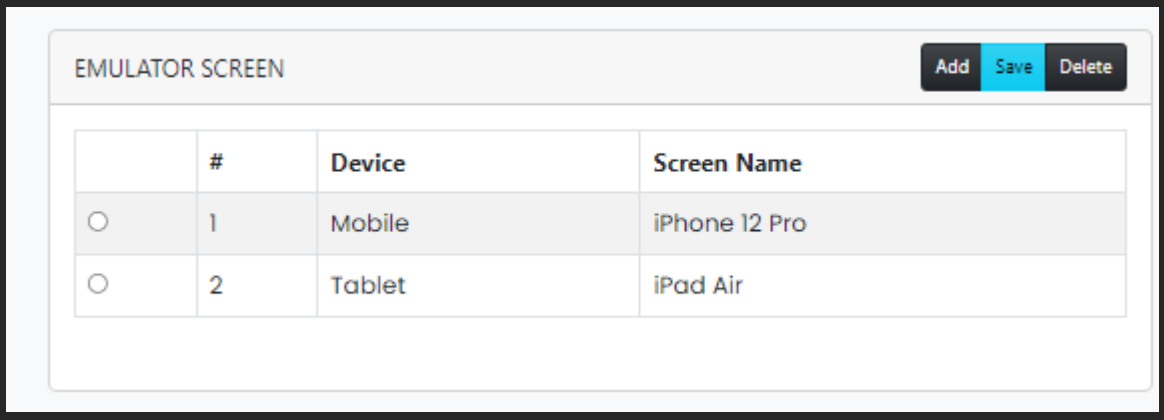
Add/Delete Emulator Screen

The emulator screen in the Chrome browser is a tool that allows you to simulate different mobile and tablet devices with various screen sizes and resolutions. By using this tool, you can test how your website or application looks and functions on different mobile devices without actually owning those devices.

To access the emulator screen in Chrome, you can open the developer tools by pressing F12 or Ctrl+Shift+I (Windows, Linux) or Cmd+Option+I (Mac), then click on the "Toggle device toolbar" icon (or press Ctrl+Shift+M) to open the device mode. From there, you can choose different device presets or customize your own by adjusting the screen size, pixel density, and user agent string.

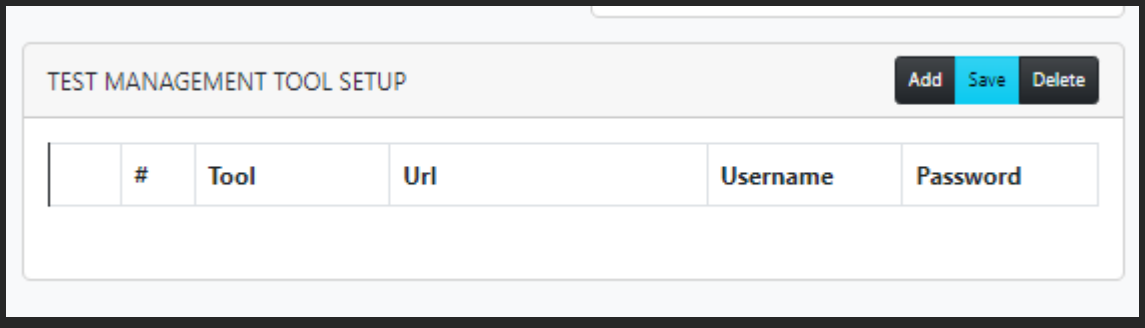
However, note that the emulator screen is just a simulation and may not fully replicate the behavior of a real mobile device. It's still recommended to test your website or application on actual mobile devices as well.

** Click on Add button select Device type and set Screen name and click Save button



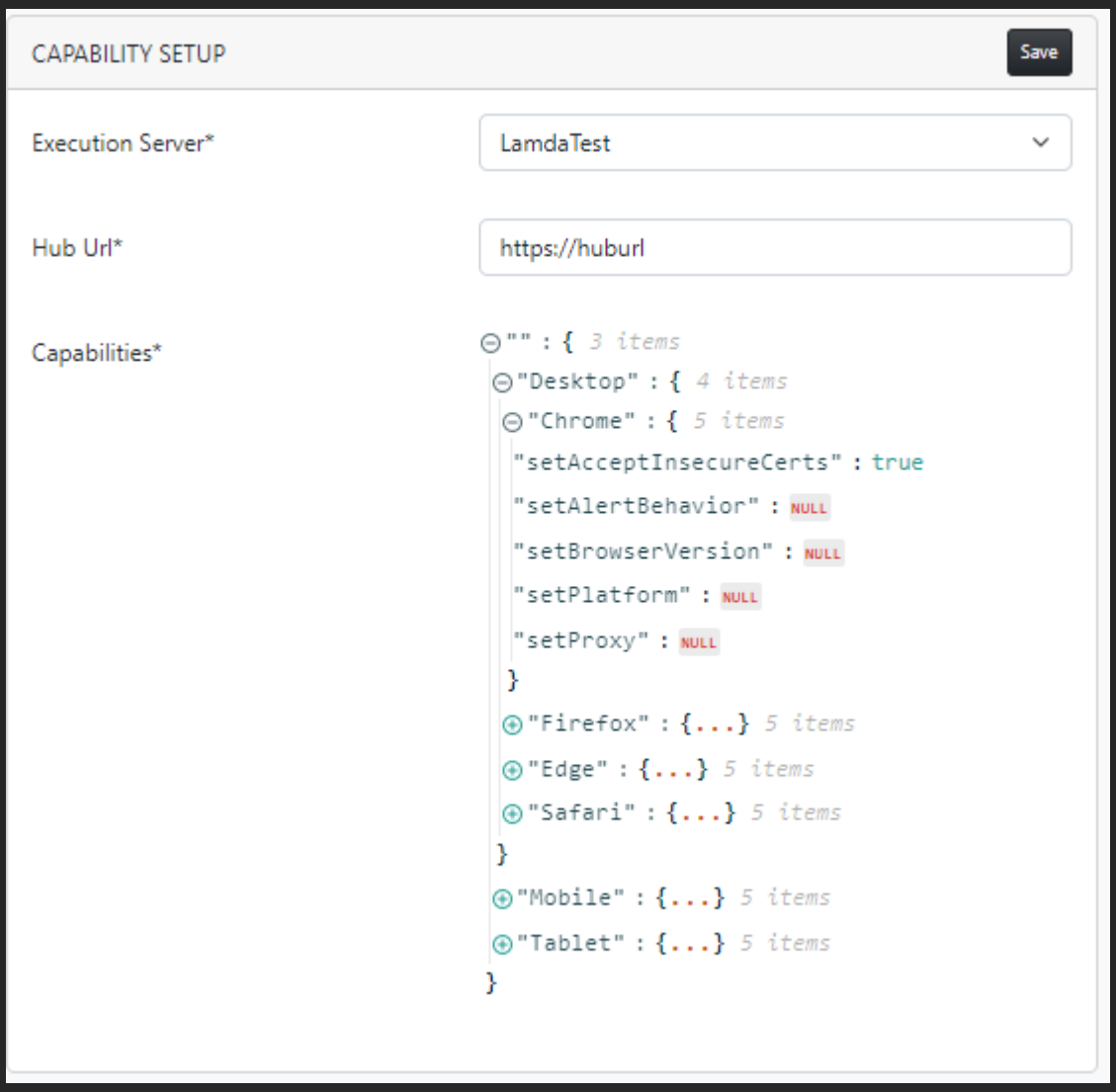
TEST MANAGEMENT TOOL SETUP

This section will used for future prospective, if client has external test management tool then we can integrate with QAautoMATER.



CAPABILITY SETUP:

You can configure the desired capabilities for different execution platforms such as desktop, mobile, and tablet for Selenium grid or other cloud provider server by updating the Hub URL and setting the desired capabilities for each platform. These capabilities can include the operating system, browser version, maximum parallel instances, and other settings that are required for the setup of the test environment. By configuring the desired capabilities, you can ensure that your tests run on the correct platforms with the required settings, which can help improve the reliability and efficiency of your test execution.


CAPABILITY SETUP
Save
Execution Server*
LamdaTest
Hub Url*
https://huburl
Capabilities*
"" : { 3 items
"Desktop" : { 4 items
"Chrome" : { 5 items
"setAcceptInsecureCerts" : true
"setAlertBehavior" : NULL
"setBrowserVersion" : NULL
"setPlatform" : NULL
"setProxy" : NULL
}
"Firefox" : {...} 5 items
"Edge" : {...} 5 items
"Safari" : {...} 5 items
}
"Mobile" : {...} 5 items
"Tablet" : {...} 5 items
}

2.2 AUTOMATION – HOW TO CREATE AUTOMATION TEST SCRIPT

While creating new test script please focus on given below steps

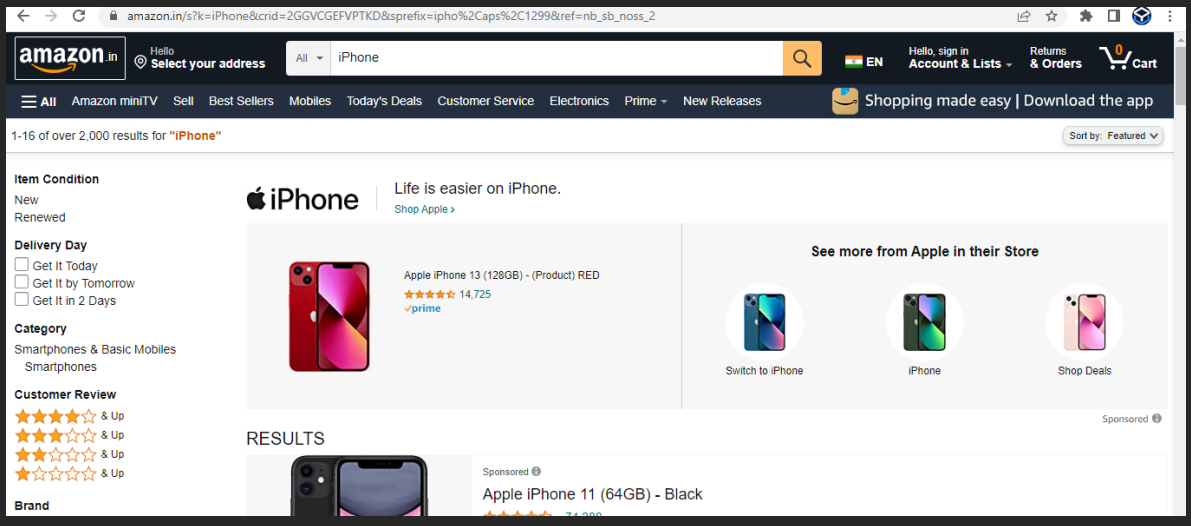
1. Identifying reusable steps involves looking at the overall testing process and identifying any steps that can be reused across multiple test cases. This can help to save time and effort in creating new test cases and can ensure consistency in testing.

2. Identify the page or feature for which test script will written.

For example, I need to write a 2 test case for the Search Result page in the Amazon application

TC-1 – for valid search result count should be greater than 0

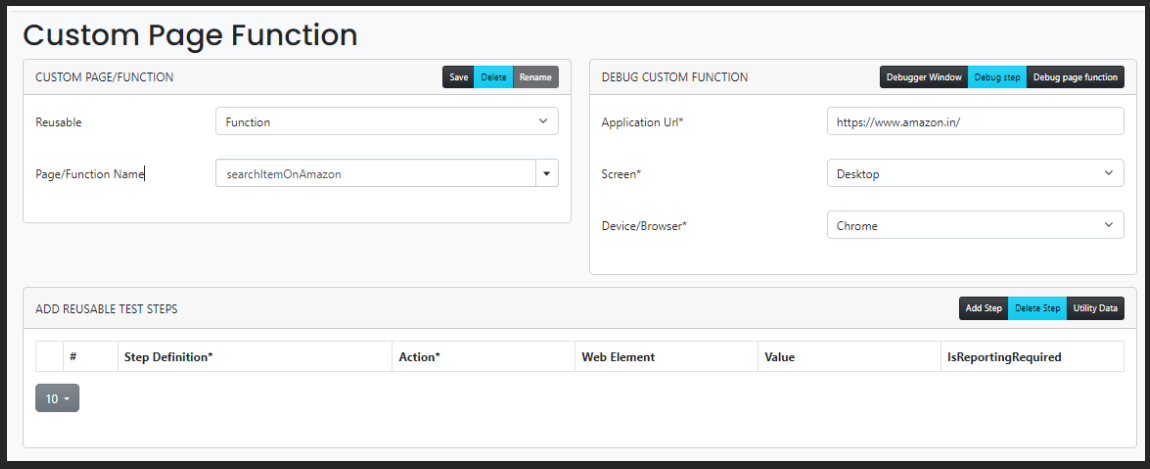
TC-2 – For invalid test case result count should be zero



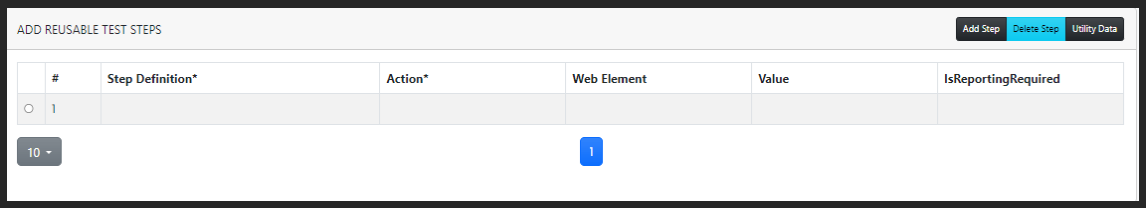
For instance, if we create a reusable method called searchItemOnAmazon, we can use it to test the Search Result page in the Amazon application across different test scripts, making our testing process more efficient and effective

Reusable Function:

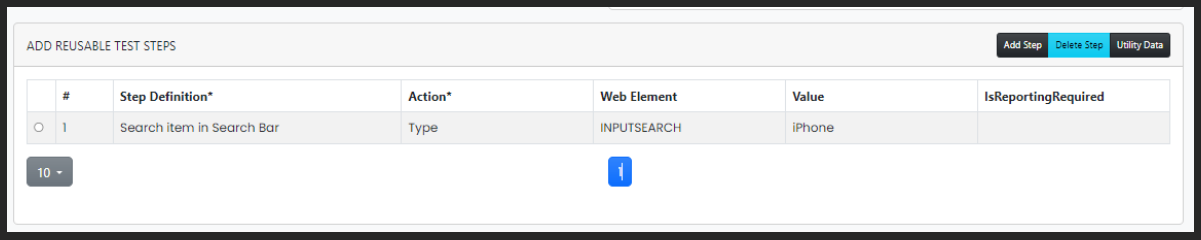
a. Navigate to Custom Page Function through navigation Automation-> UI Testing > Custom Page Function.
b. Select Function from Reusable dropdown and add your method name, here I am adding method 'searchItemOnAmazon'



c. Click on Add Step button

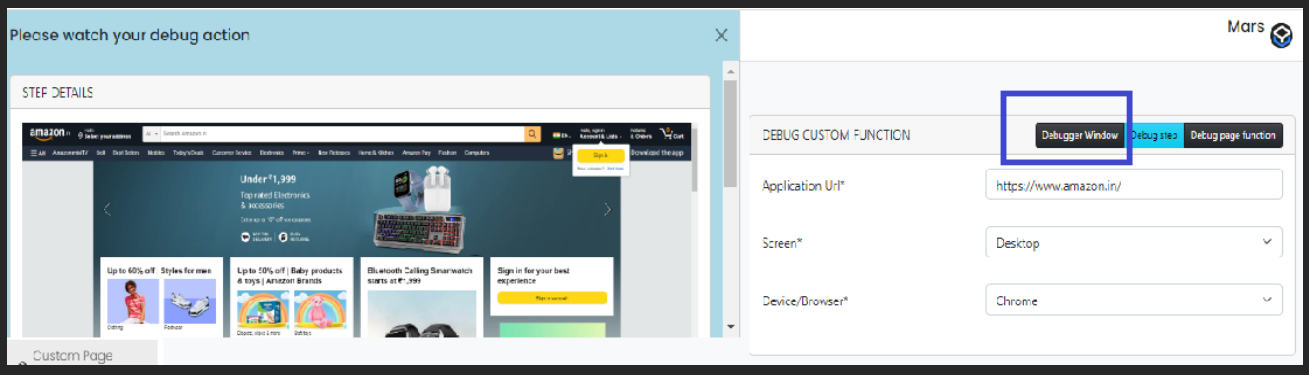


d. Add Step definition , here I am writing 'Search item in Search Bar' , once you done with Step definition write 80-90% times Action and web element will Auto generated.

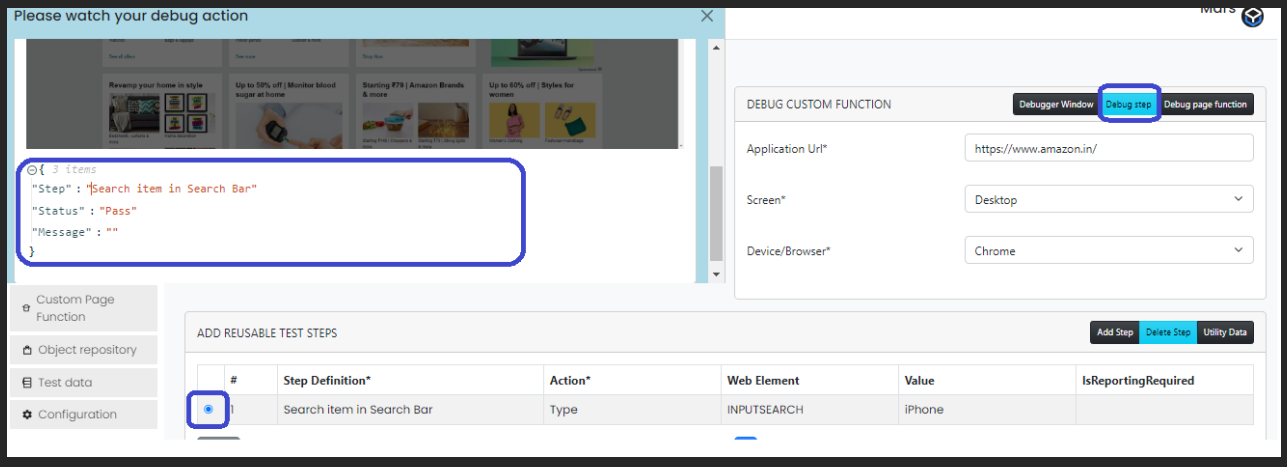


e. In value section, provide any value what we want to provide search item.

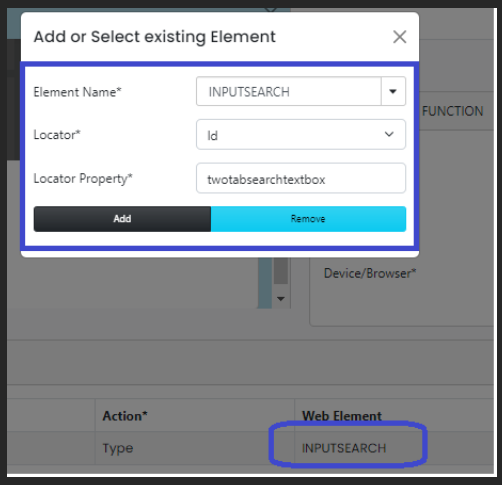
f. To make sure your automated step is working fine , click on the Debugger window from the Debug Custom Function section



g. Select the radio button for test step and Click Debug step button

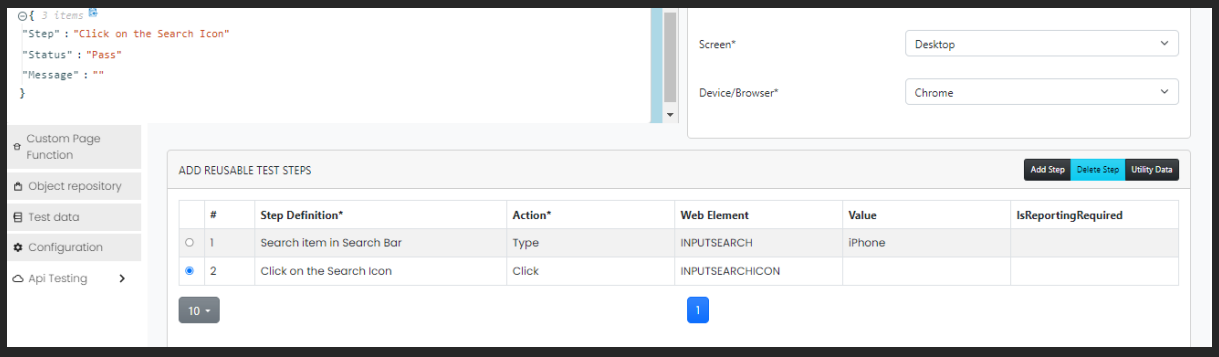


h. If you are getting Status Fail then update the Locator, Click the Web Element column from selected Row , and update locator property . After updating the Property Click Debug Step and make sure status is PASS



i. Parameter the test data , in value section write ARGS.searchItem (ARGS. Is the mandatory keyword to parameterizing the value)

j. Similarly we can write different steps , here we are writing another steps 'Click on the Search Icon'



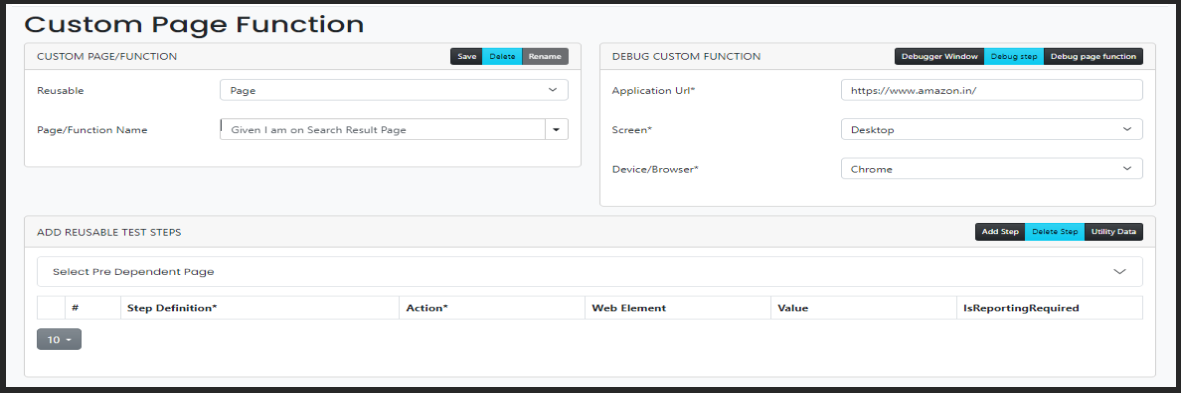
h. Close the debugger window and Click on the Save button

Coming to our example scenario 2 scenario are for search result page so if we can create Page function for search Result page so every test script can consume same page function, which is written for search result page.

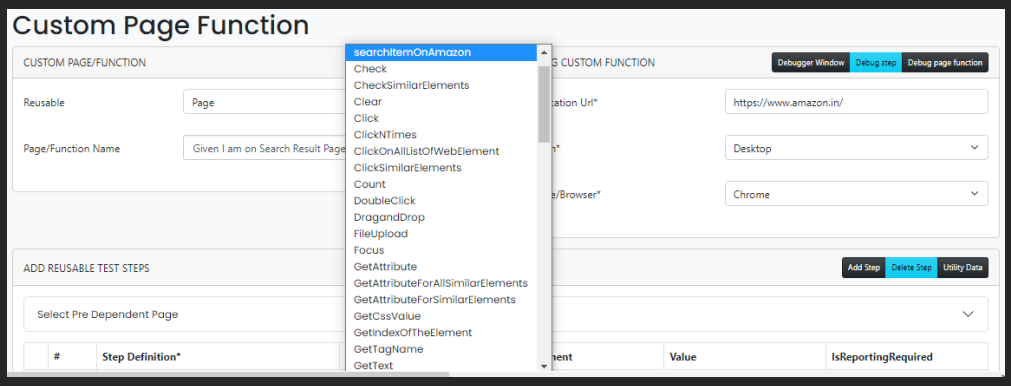
Page Function:

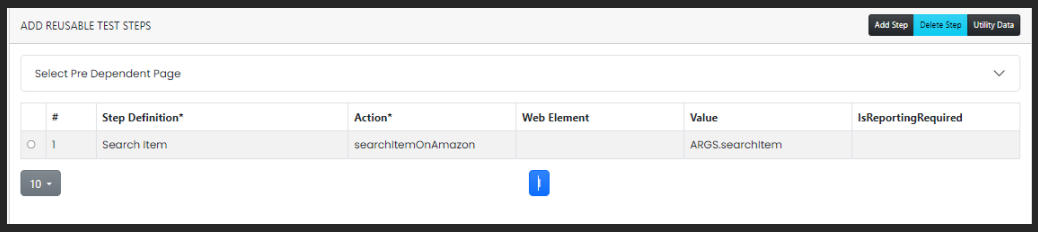
a. Navigate to Custom Page Function through navigation Automation-> UI Testing > Custom Page Function.

b. Select Page from Reusable dropdown and add your page name, here I am adding page name 'Given I am on Search Result Page'



c. Click on the Add Step button and click on Action column and select newly created Reusable method





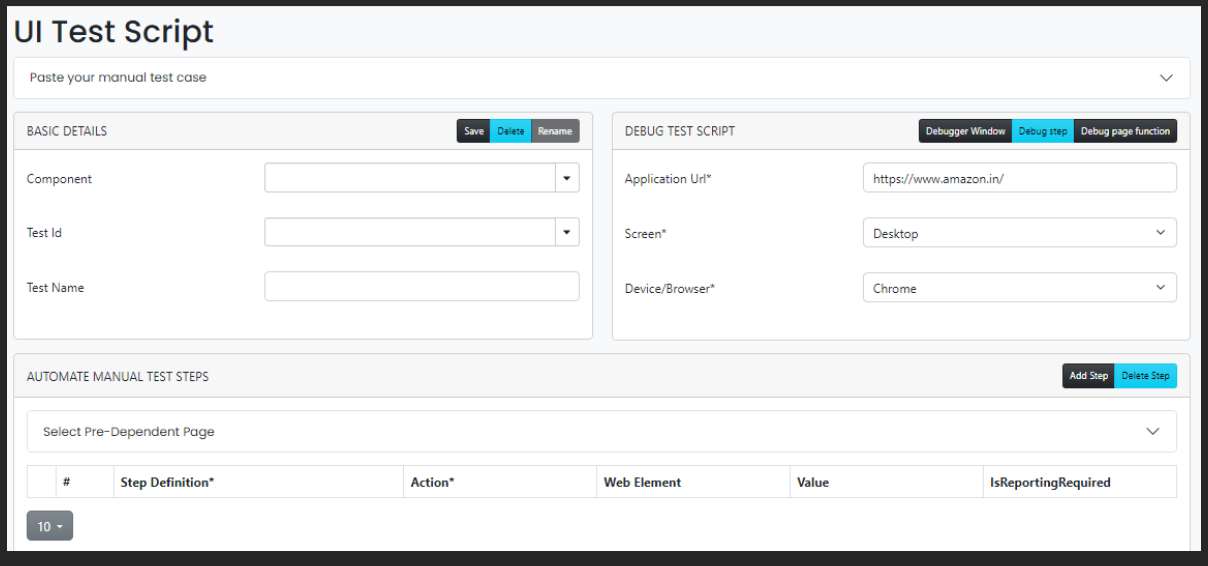
d. Add more step if required to reaching respective page, also debug your step to make sure your page function is also working.

e. Click on Save button

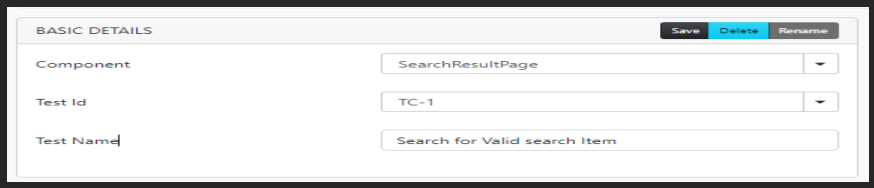
New Test Script:

After creation of reusable method and page function

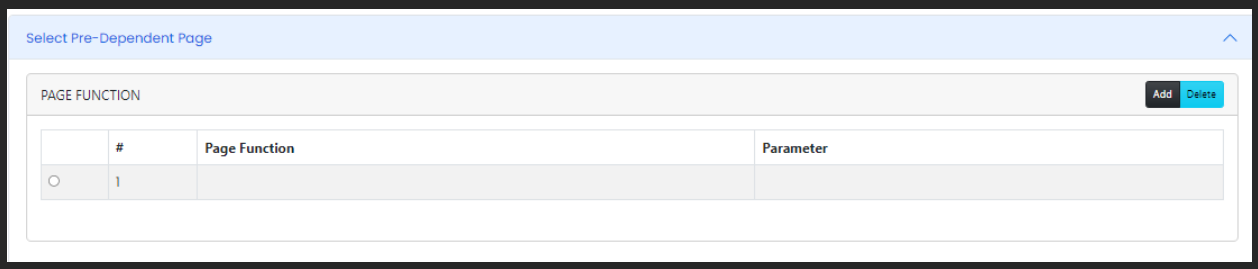
a. Navigate to UI Test script page through navigation Automation-> UI Testing > Test Script



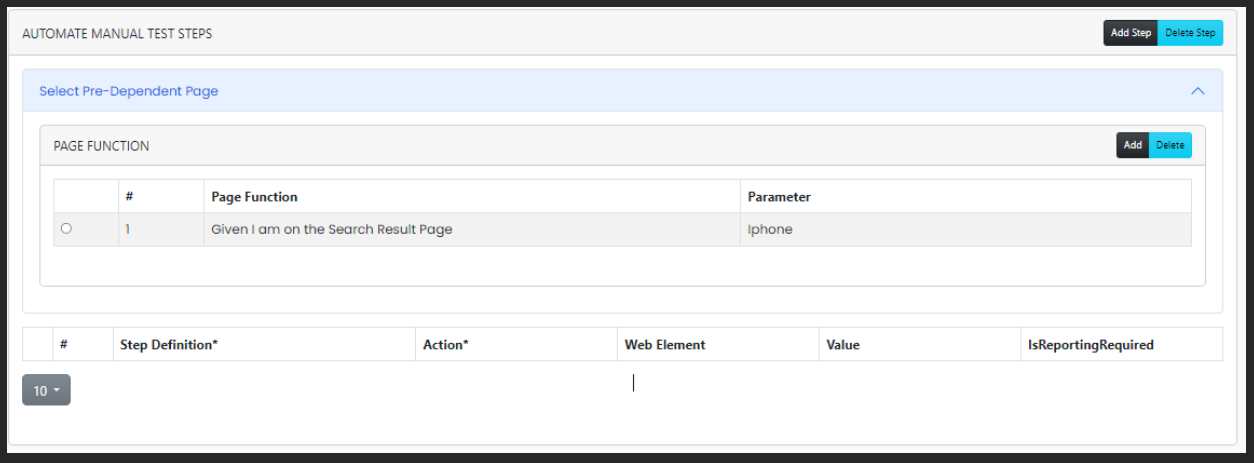
b. Provide basic details Component , Test ID and Test name from the basic details section



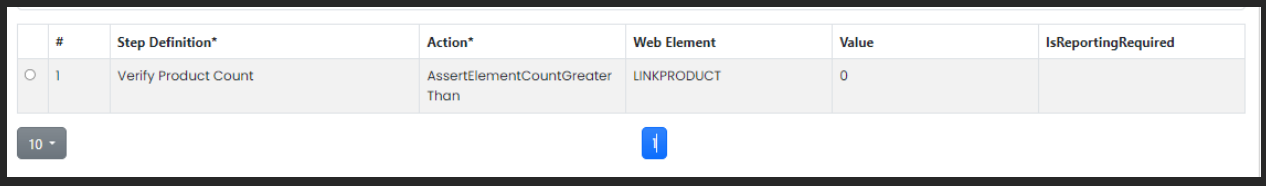
c. Expand Pre Dependent Page section and click Add button



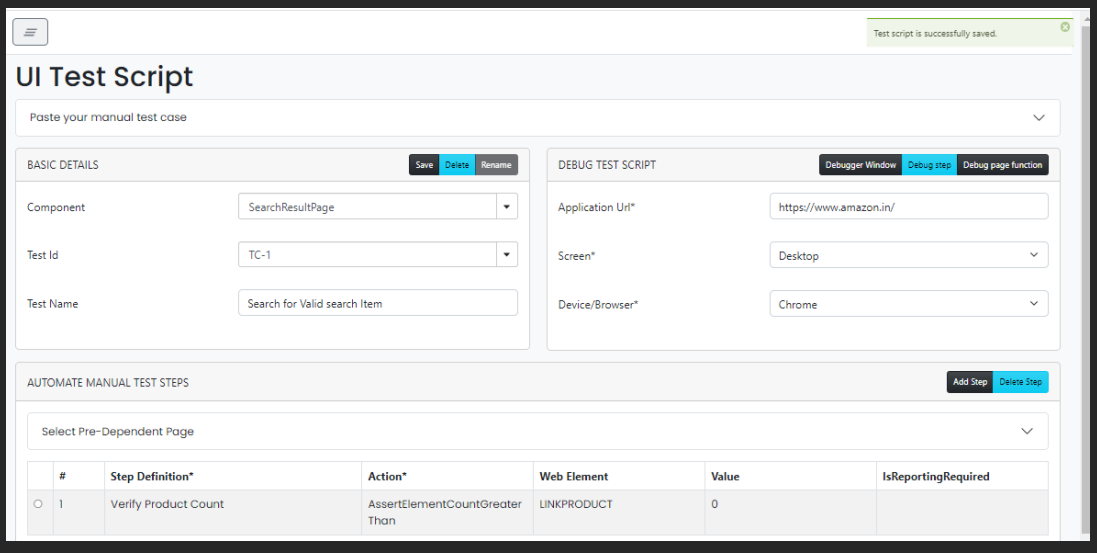
d. Select existing Page function and update the Parameter value



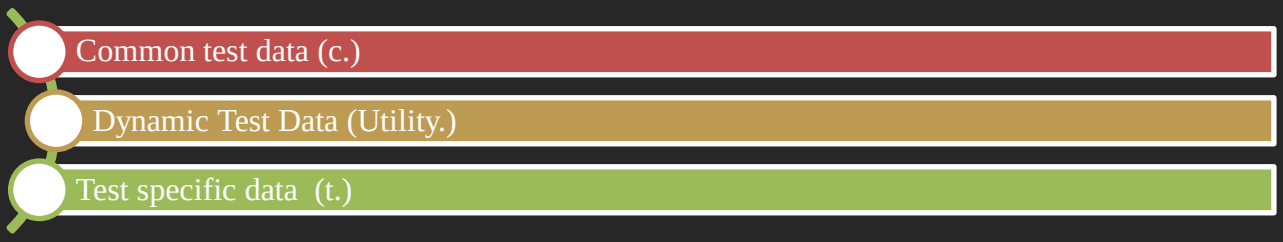
e. Now click on Add step button , Add step definition , select Appropriate action , element and Value



f. Create all step in similar way make sure all step is working find by checking debug functionality and save the test script.

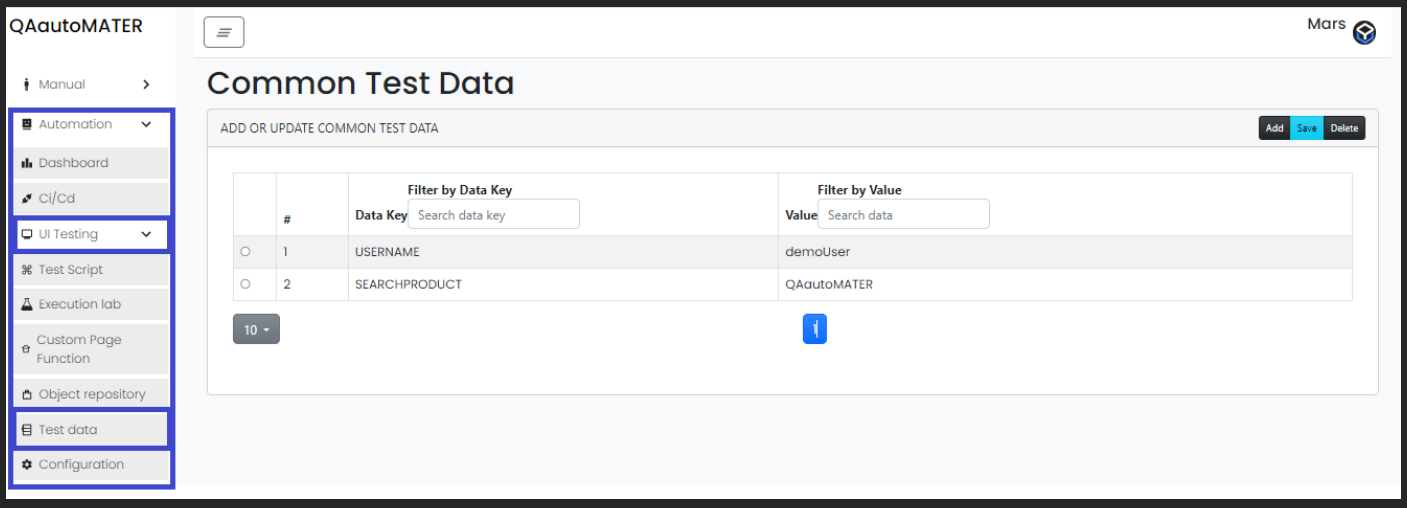


2.3 AUTOMATION – TEST DATA MANAGMENT



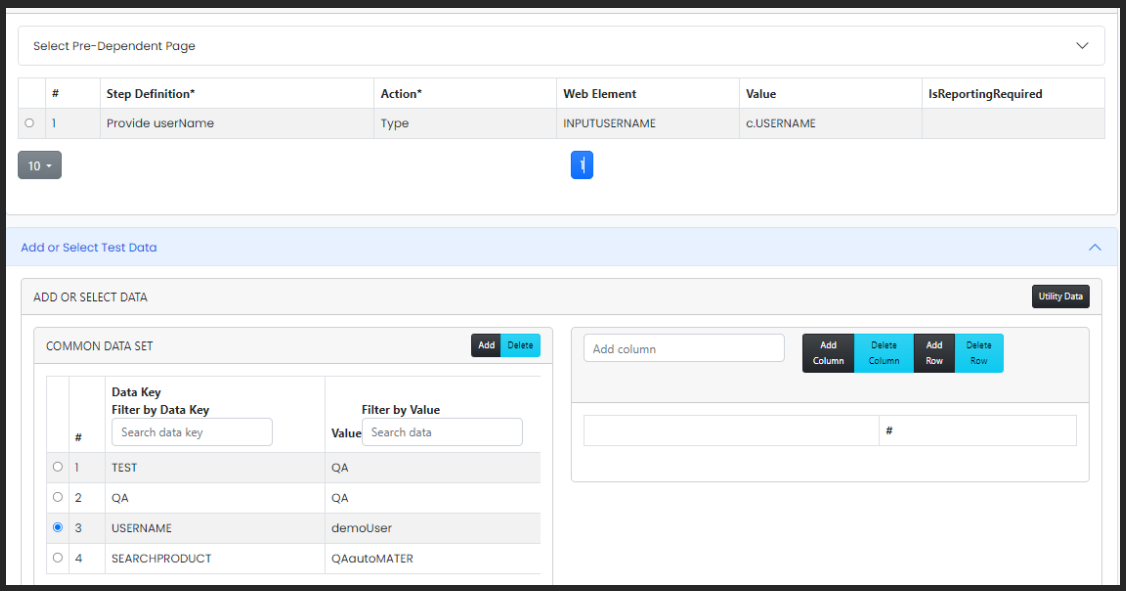
Common Test Data:

When multiple test scripts use the same test data, it is a good practice to add this data to the 'Test Data' page under the 'UI Testing' section.

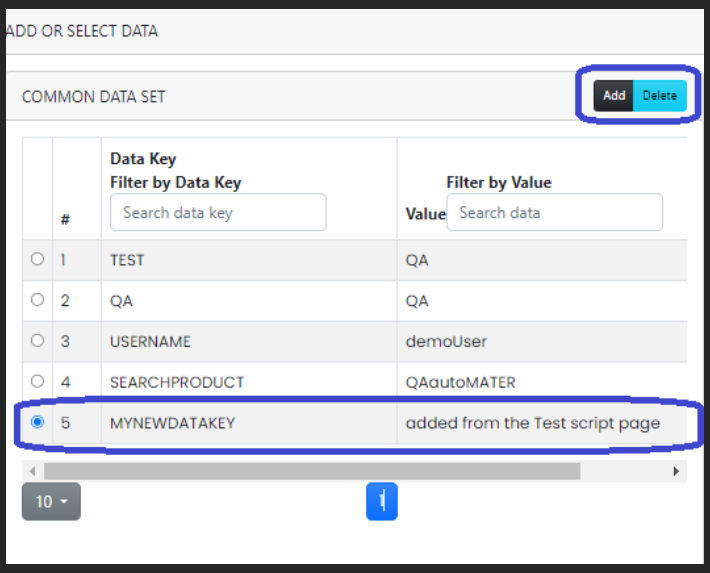


How to Consume test data in Test script/Custom Function:

After adding the common test data, this test data can be directly use in test Scripts and custom Page function by calling "c.dataKeyName" . Here is the example c.USERNAME



From the UI test script page, new Common test data can also be added from the ADD OR SELECT DATA section.

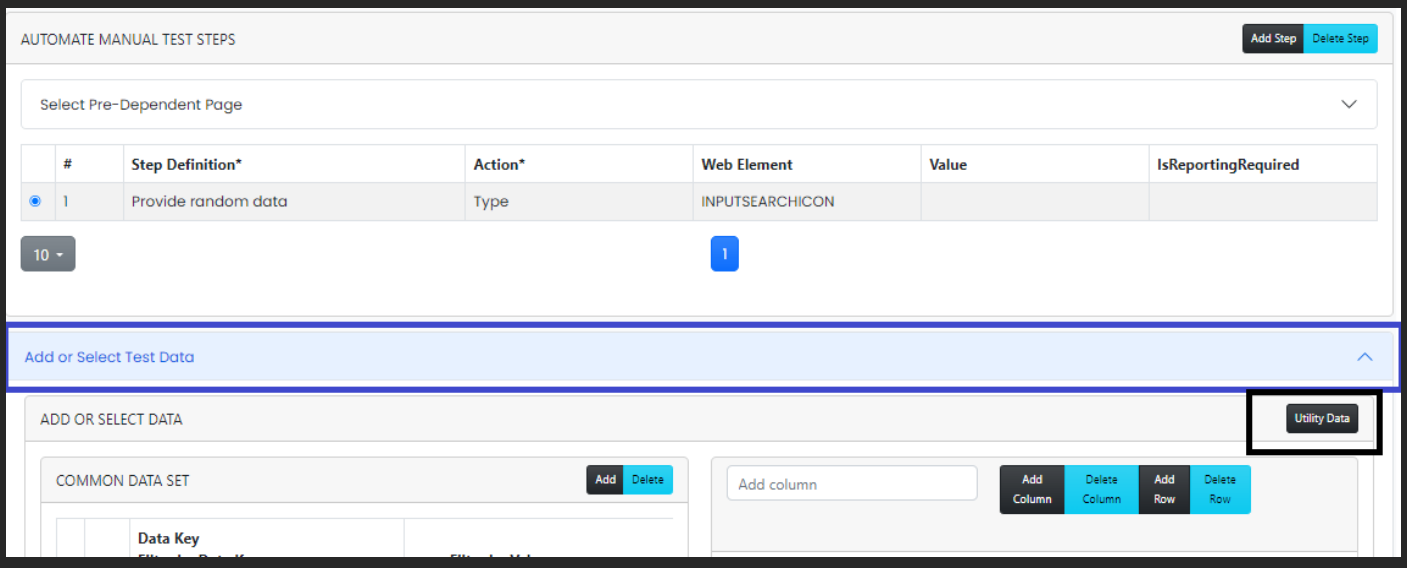


Dynamic Data:

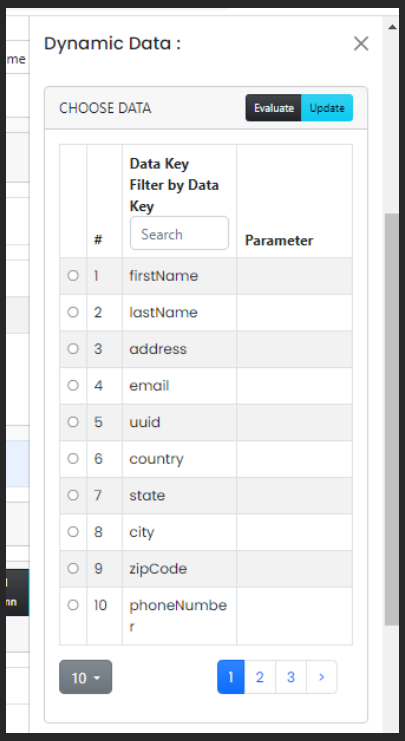
QAAutoMATER supports dynamic test data, such as emails, random strings, and phone numbers. You can generate this test data and use it in your test scripts or custom function pages, as well as for arithmetic operations such as addition, subtraction, multiplication, and percentage calculation. Additionally, you can use string functions such as splitting, among others.

Example:

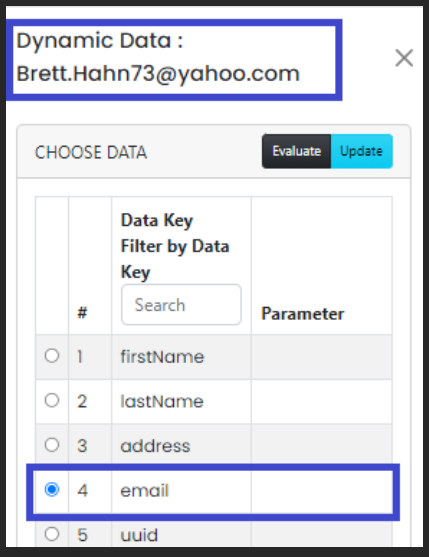
Navigate to Test Script page or Custom Function page and Expand Add or Select Test Data



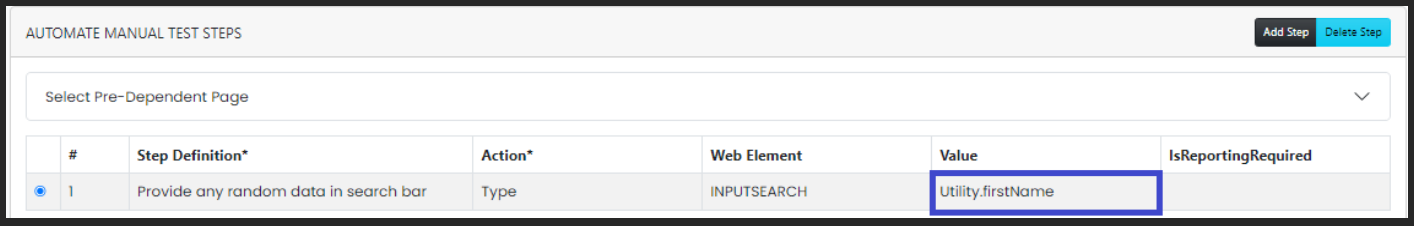
Select test step and click Utility Data button



Select data key as per test data requirement and click Evaluate button to make sure data is generated in correct way e.g.



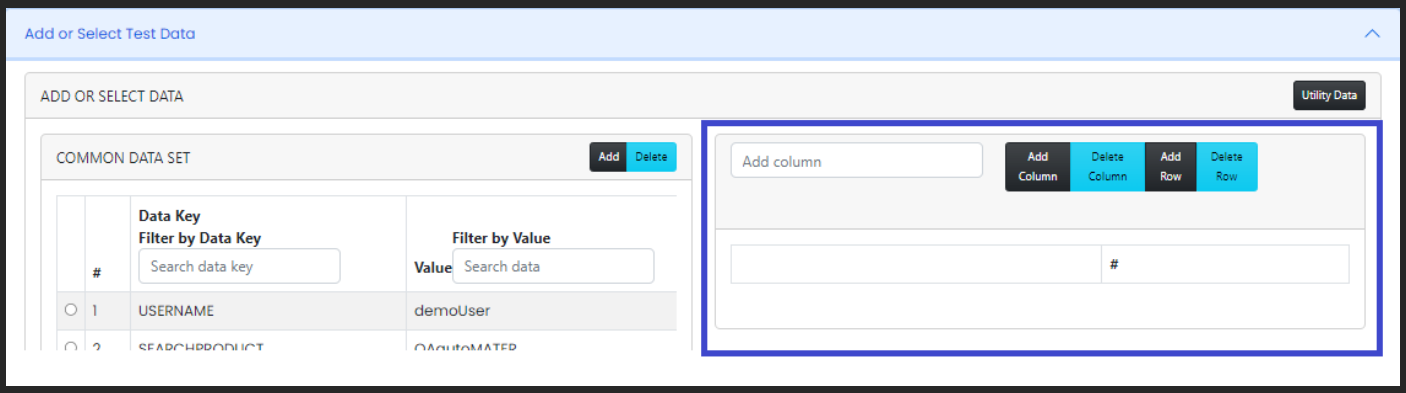
After successful evaluation Click on the Update button, Dynamic data will be added in test step, starting with Utility.



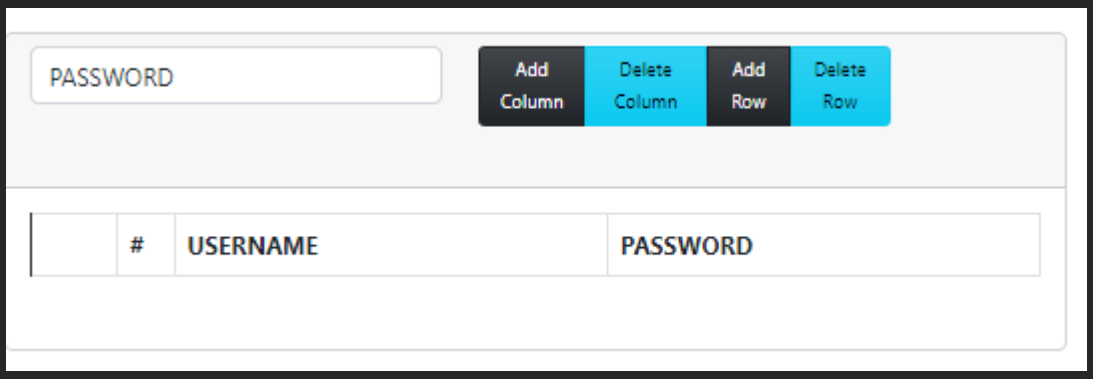
Please Note : you can add multiple dynamic data in single step e.g. Utility.firstname+Utility.lastName+@qabunch.com

Test Specific Data:

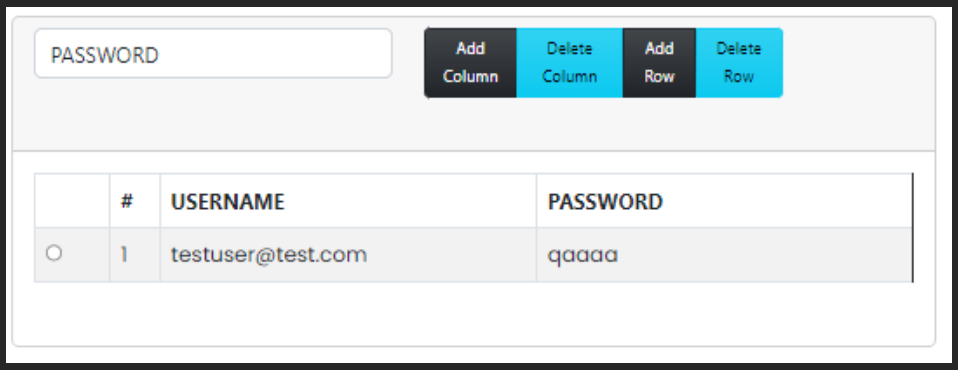
Test data, which is only applicable for Test script, such kind of test data can be added from the test script page. On the UI test script page, expand the section 'Add or Select Test data' section.



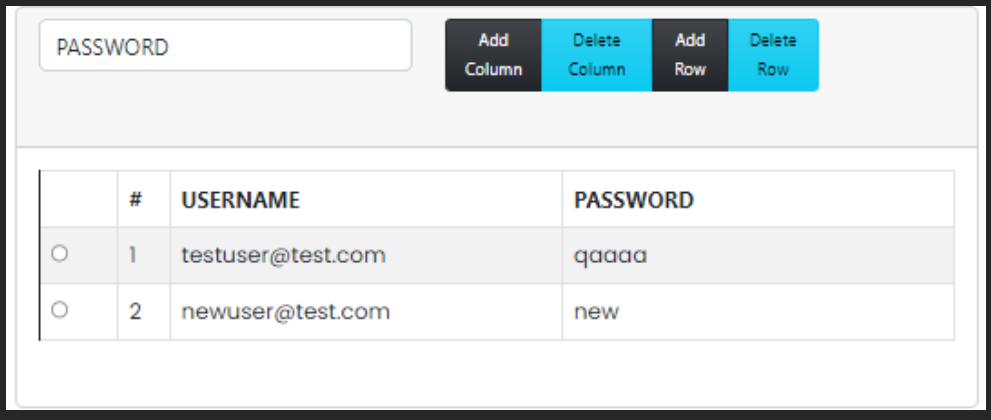
Provide Column or test data logical name in Add column Edit box and click Add Column button



Click on Add Row button and Provide Test data value



If you want to run your test script on multiple iteration, add another row with data.



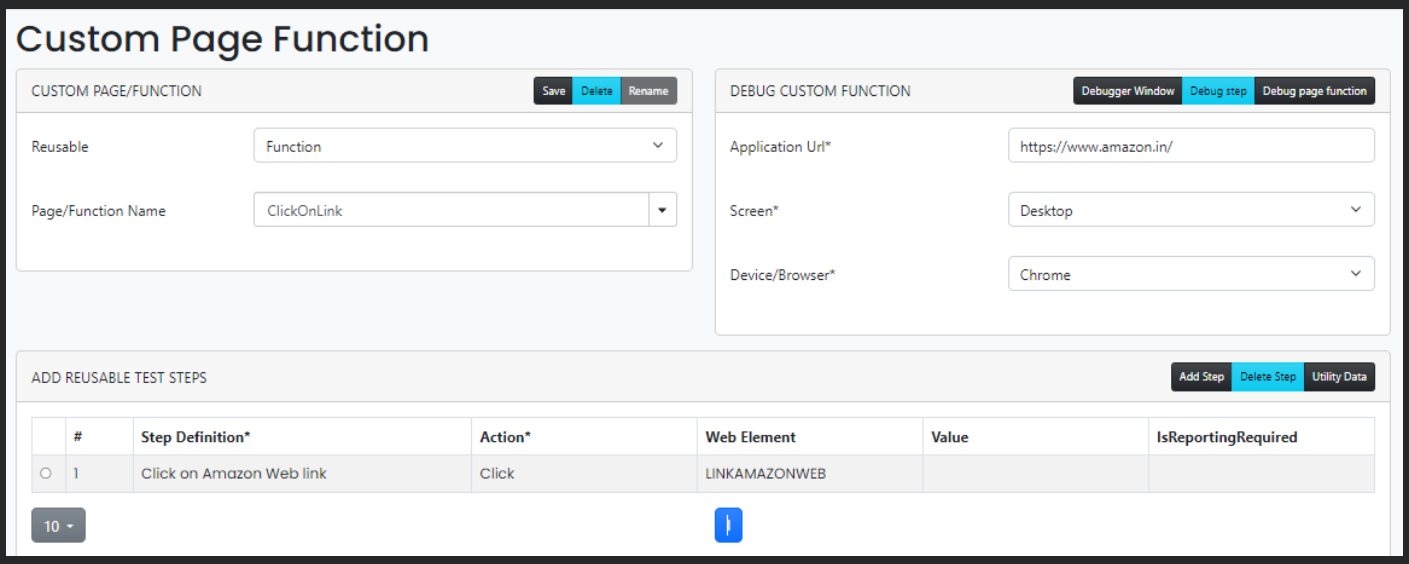
2.4 OR – LOCATOR PARAMETERIZATION

Locator property can be parameterize using test data (c., t.Utility.) , argument data (ARGS.). and Runtime data (Session.)

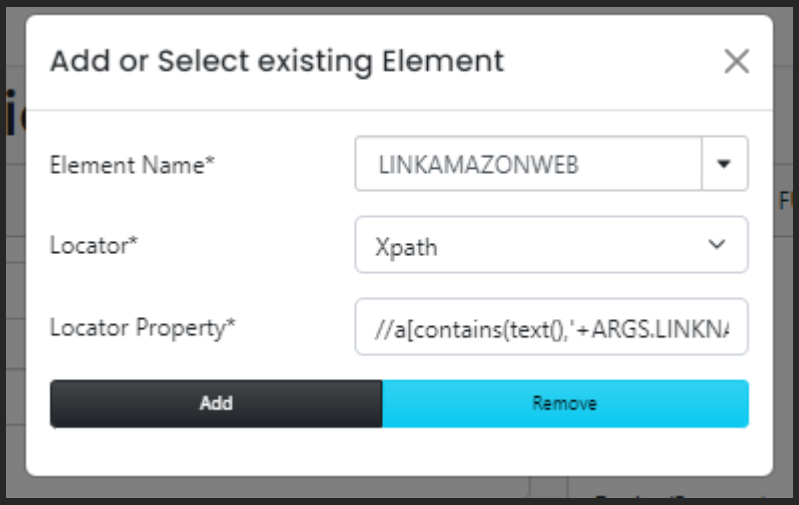
Example:

Scenario 1 : In the Amazon application, there are many web links, and you want to click on a link without having to add a locator for each one. You would like to simply provide the text of the link, and have the web element created automatically.

Solution: Go to the Custom Page function, add any reusable method name and click Add Step button



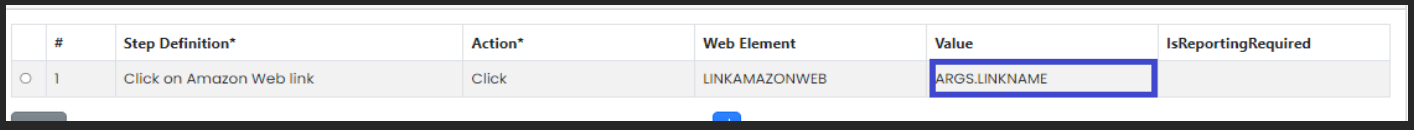
Click on Web Element Column for Step



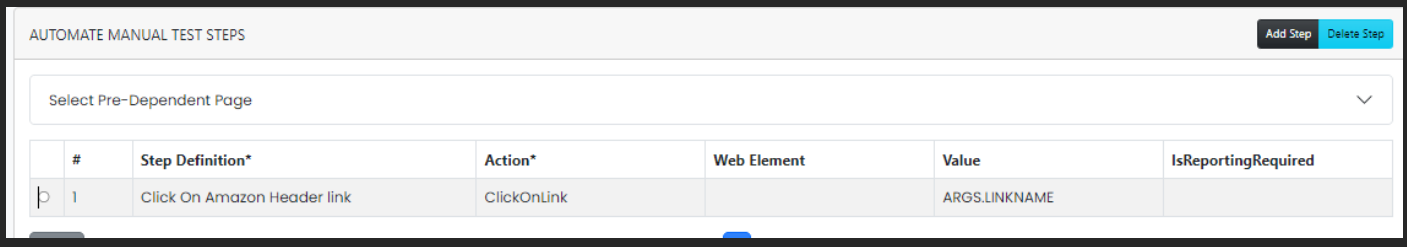
Parameterize the Xpath so here xpath is //a[contains(text(),'+ARGS.LINKNAME+')] ,

- Parameter will always started with + and end with +

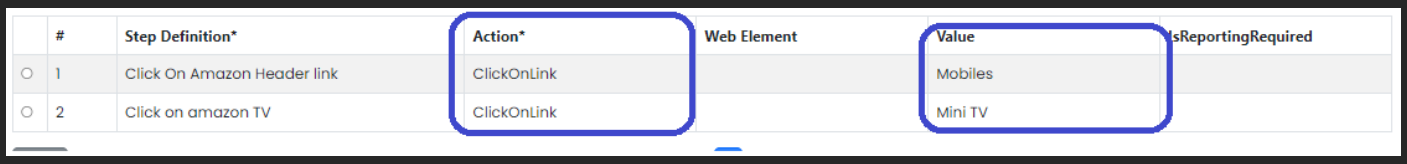
Click on Add button and value selection Provide the Same Name which is used in Xpath ARGS.LINKNAME



After successful creation of reusable method, go to the test script page and select this reusable method from action dropdown

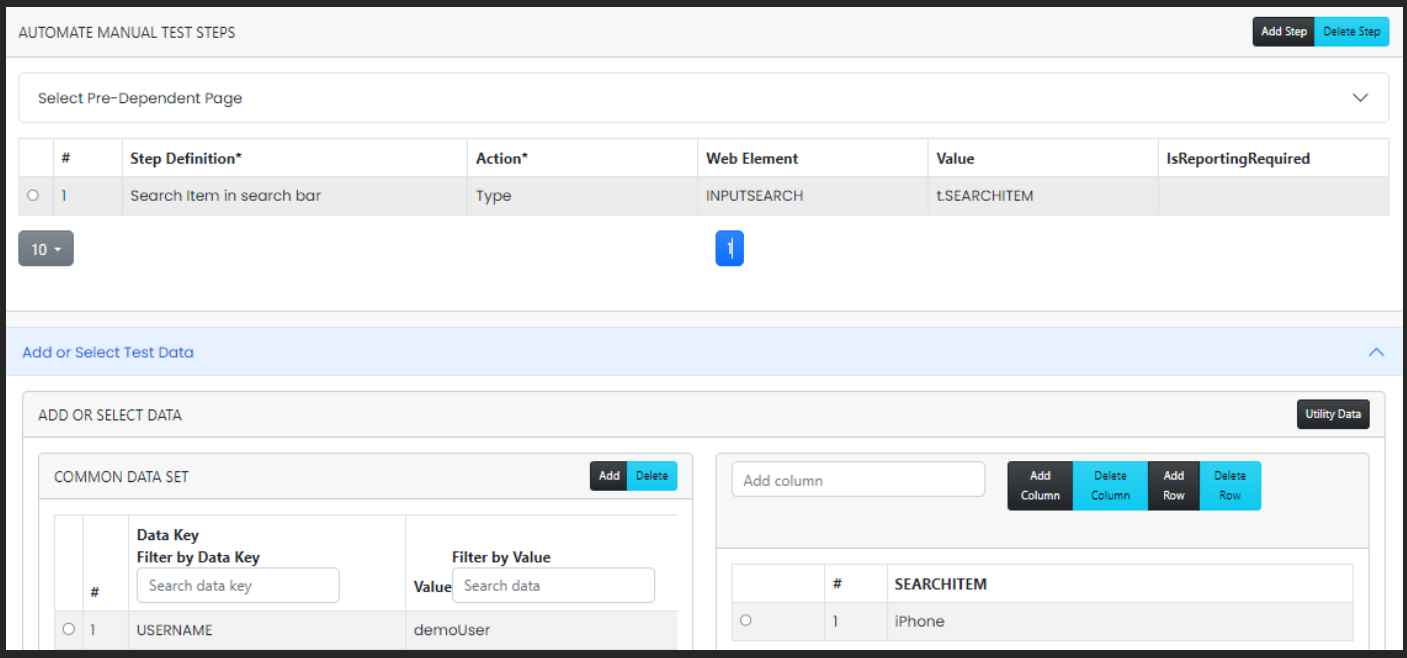


Now call this reusable method for different link



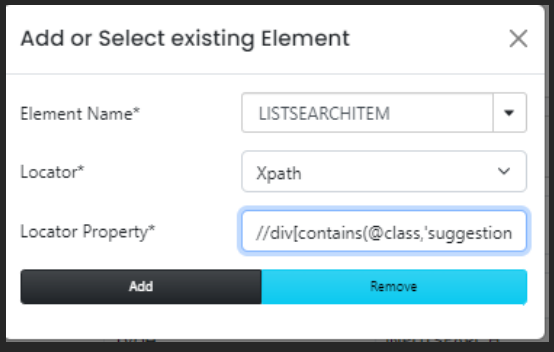
Scenario 2: When you type something in the Amazon search bar, the auto-populated item with the same text must be clicked

Solution: Go to the Test script page Perform search action and provide test data



Add Step for Auto populated selection and click Web element column and parameterize the Xpath

//div[contains(@class,'suggestion')]//div[contains(text(),'+t.SEARCHITEM+')]



2.5 SAVE SESSION VARIABLE OR HANDLE DATA ON RUN TIMES

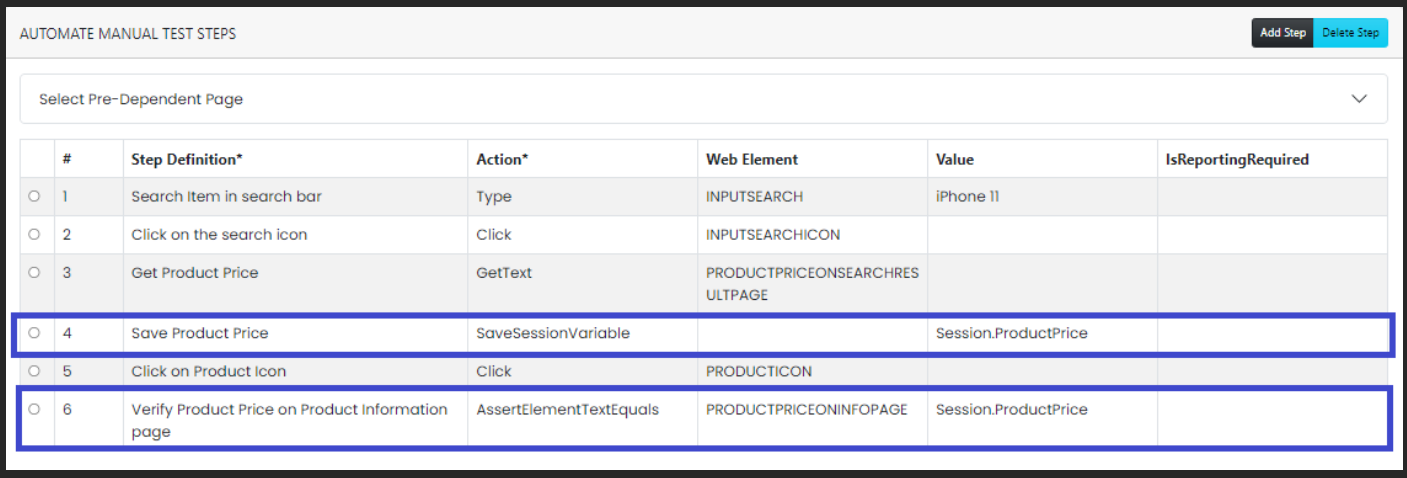
QAautoMATER provides support for capturing runtime data, saving it, and using it across different steps. This data can also be used to parameterize locator properties.

Example:

Scenario 1 : In the Amazon web application, you have to get product price on search result page and verify same product price on product information page

Solution:

Action "SaveSessionVariable" which captures the value of the step that was just executed and saves it in a variable with a specified name



	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
○	1	Search Item in search bar	Type	INPUTSEARCH	iPhone 11	
○	2	Click on the search icon	Click	INPUTSEARCHICON		
○	3	Get Product Price	GetText	PRODUCTPRICEONSEARCHRESULTPAGE		
○	4	Save Product Price	SaveSessionVariable		Session.ProductPrice	
○	5	Click on Product Icon	Click	PRODUCTICON		
○	6	Verify Product Price on Product Information page	AssertElementTextEquals	PRODUCTPRICEONINFOPAGE	Session.ProductPrice	

Please Note: the scope of variable is on test scripts only, you cannot use this variable on another scripts.

2.1 DYNAMIC ACTION

QAAutoMATER tool supports to perform action on dynamic web element

Scenario 1.

You want to perform based on index

Example

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
○	1	click on 6 th item of element	ClickSimilarElements	LISTOFPRODUCT	{"index":5}	
◉	2	Click on 6th item of the element	ClickSimilarElements	LISTOFPRODUCT	index=5	

Scenario 2.

You want to type on edit box , if similar element is present

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
○	1	Provide information for guest	TypeSimilarElements	LISTOFPRODUCT	{"src":"qaBunch"}	
◉	2	Provide information for guest	ClickSimilarElements	LISTOFPRODUCT	src=qaBunch	

Scenario 3 :

You want click on search result item which text is something

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
○	1	Click on search Item Link	ClickSimilarElements	LISTOFPRODUCT	{"text":"qaBunch"}	
◉	2	Click On Search Item Link	ClickSimilarElements	LISTOFPRODUCT	text=qaBunch	